
900 Series HP 3000 Computer Systems
MPE/iX Shell and Utilities
Reference Manual

Volume 1

**HEWLETT
PACKARD**

HP Part No. 36431-90001
Printed in U.S.A. 1994

Second Edition
E0494

MKS and InterOpen are trademarks of Mortice Kern Systems Inc.

UNIX is a registered trademark of Unix Systems Laboratories, Inc. in the U.S.A. and other countries.

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Copyright © 1992,1994 by Mortice Kern Systems Inc. and Hewlett-Packard Company

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013. Rights for non-DOD U.S. Government Departments and Agencies are set forth in FAR 52.227-19(c)(1,2).

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Printing History

The following table lists the printings of this document, together with the respective release dates for each edition. The software version indicates the version of the software product at the time this document was issued. Many product releases do not require changes to the document. Therefore, do not expect a one-to-one correspondence between product releases and document editions.

Edition	Date	Software Version
First Edition	October 1992	A.00.00
Second Edition	???	???

Preface

MPE/iX, Multiprogramming Executive with Integrated POSIX, is the latest in a series of forward-compatible operating systems for the HP 3000 line of computers.

In HP documentation and in talking with HP 3000 users, you will encounter references to MPE XL, the direct predecessor of MPE/iX is a superset of MPE XL. All programs written for MPE XL will run without change under MPE/iX. You can continue to use MPE XL system documentation, although it may not refer to features added to the operating system to support POSIX (for example, hierarchical directories).

Finally, you may encounter references to MPE V, which is the operating system for HP 3000s, not based on PA-RISC architecture. MPE V software can be run on the PA-RISC (Series 900) HP 3000s in what is known as *compatibility mode*.

The first edition of the *MPE/iX Shell and Utilities Reference Manual Volumes 1 and 2* (customer order number 36431-60001) is a two volume manual set that provides reference descriptions of the commands and utilities available through the MPE/iX Shell.

- Volume 1 provides MPE/iX Shell command and utility descriptions alphabetically from A through M.
- Volume 2 provides command and utility descriptions alphabetically from N through Z.

The *MPE/iX Shell and Utilities Reference Manual Volumes 1 and 2* is organized into five chapters and one appendix. In addition, both volumes contain a common table of contents, permuted index, and index that provide cross references into both volumes.

Volume 1

Chapter 1 **Commands and Utilities** contains MPE/iX Shell command and utility descriptions alphabetically from A through M.

Volume 2

Chapter 1 **Commands and Utilities** provides MPE/iX Shell command and utility descriptions alphabetically from N through Z. (This chapter is a continuation of the Chapter 1 found in Volume 1.)

Chapter 2 **File Formats** provides more detailed information on the formats of files used by the various commands and utilities described in chapter 1.

Chapter 3

Miscellaneous Information provides details on miscellaneous topics not covered in other chapters.

Appendix A

MPE/iX Implementation Considerations provides an overview of implementation considerations you need to understand when using MPE/iX Shell and Utilities on a 900 Series HP 3000 computer system.

Conventions

Throughout this manual, the following conventions help you to distinguish between different elements of text and to learn about MPE/iX Shell and Utilities.

Convention	Description
<code>courier</code>	Literal user input, directory names, file names, and path names are expressed in normal Courier font (that is, typewriter font).
bold courier	Commands and command line options in a synopsis line or embedded in regular text are in bold Courier font.
[]	Optional command line items and optional parts of command names are enclosed in square brackets. For example, [-z] indicates that the -z option can be specified, but is not required.
	This manual uses the or-bar () to indicate a mutually exclusive choice of command line items. For example, -a -b indicates that you can specify either the -a option or the -b option, but not both.
...	In a command synopsis, the ellipsis indicates that a command line item can be repeated any number of times. In examples, it indicates where portions of an example have been omitted.
bold	Commands internal to interactive utilities are shown in bold Roman font (for example, ZZ in vi, and followup in mailx).
<i>ITALIC COURIER</i>	Environment variables are expressed in uppercase italic Courier font.
<i>italics</i>	Placeholders which are to be replaced when actually entering a command are shown in italics. For example, you would replace <i>filename</i> with an actual file name.
SMALL CAPS	Acronyms and combination key sequences are indicated with small caps. When you see the - sign between two key names, such as ALT or CTRL, hold down the first key while pressing the second. For example, to enter ALT-X, hold down the ALT key and press X.
ENTER	This manual uses ENTER to stand for the key that is sometimes labelled RETURN, or has an arrow pointing down and to the left.

Table of Contents

Volume 1

Chapter 1. Commands and Utilities

intro	introduction to man pages	1-2
alias	display or create command aliases	1-8
ar	create and maintain library archives	1-11
asa	interpret ASA/FORTRAN carriage control	1-15
awk	data transformation, report generation language	1-17
banner	display text in large font	1-42
basename	display file name component of path name	1-44
bc	arbitrary-precision arithmetic calculation language	1-45
break	exit from loop in shell script	1-68
c	produce multi-column output	1-69
c89	generic C compiler interface	1-71
calendar	appointment reminder system	1-76
callci	run a MPE/iX CI command from the MPE/iX Shell	1-79
cat	concatenate and display text files	1-81
cd	change working directory	1-83
chgrp	change the group ownership of files and/or directories	1-86
chmod	change access permissions of a file	1-88
chown	change the ownership of files and/or directories	1-92
ci	check in a file under RCS	1-95
cksum	compute checksum and byte count for file	1-105
cmp	compare two files	1-107
co	check out a file under RCS	1-109
:(colon)	do nothing, successfully	1-118
comm	compare sorted files and show differences	1-119
command	execute a simple command	1-121
compress	Lempel-Ziv compression of a file	1-123
continue	skip to next iteration of loop in shell script	1-127
cp	copy files	1-128
cpio	archiver to copy and back up files	1-133
csplit	split a text file, according to criteria	1-138
ctags	produce tags file for ex , more , and vi	1-143
cut	selectively display fields or characters from input lines	1-147
date	set and display date and time	1-150
dc	arbitrary precision desk calculator	1-155
dd	copy and convert input blocks	1-168

diff.....	compare two text files and show differences	1-174
diffh.....	compare two text files and show differences	1-174
bdiff.....	compare two text files and show differences	1-174
diff3.....	compare three text files	1-181
diffb.....	compare binary files and show differences	1-185
dirname.....	display directory components of path name	1-187
.(dot).....	execute shell file in current environment	1-189
du.....	summarize file space usage	1-191
echo.....	display arguments	1-193
ed.....	line-oriented text editor	1-195
red.....	line-oriented text editor	1-195
env.....	display environment, set environment for process	1-207
eval.....	evaluate arguments in shell	1-209
ex.....	text editor	1-211
exec.....	execute a command in place of the current shell	1-212
exit.....	exit from the shell	1-214
expand.....	expand tabs to spaces	1-215
export.....	mark names for export	1-217
expr.....	evaluate expression	1-219
false.....	fail, quietly	1-223
fc.....	display, fix, edit and re-enter previous commands	1-224
file.....	determine file type	1-227
find.....	find files within file tree	1-231
fmt.....	simple text formatter	1-238
fold.....	break lines into shorter lines	1-241
frombyte.....	convert a byte stream files to MPE record files	1-243
functions.....	display or modify shell functions	1-245
getconf.....	display POSIX configuration information	1-247
getopt.....	external command to parse shell file options	1-253
getopts.....	parse options from shell script command line	1-256
grep.....	match patterns in a file	1-259
egrep.....	match patterns in a file	1-259
fgrep.....	match patterns in a file	1-259
hash.....	create a tracked alias	1-262
head.....	display first part of file	1-263
help.....	display brief command explanations	1-265
history.....	display command history	1-267
id.....	display user and group names	1-269
ident.....	look for keywords in a file	1-271
integer.....	declare an integer variable	1-273
jobs.....	display status of jobs in current session	1-274
join.....	join two sorted, textual relational databases	1-276
kill.....	terminate process	1-280
lc.....	list file system elements in categories	1-283
let.....	evaluate arithmetic expressions	1-285

lex.....	lexical analyzer generator	1-287
line.....	copy one line of standard input	1-292
ln.....	create a link to an existing file	1-293
logname.....	display user name	1-298
ls.....	list file and directory names and attributes	1-299
mailx.....	read electronic mail	1-304
make.....	maintain program-generated and interdependent files	1-334
man.....	print sections of the online reference manual	1-362
merge.....	three-way file merge	1-367
mesg.....	allow or refuse messages	1-370
mkdir.....	create a new directory	1-372
mkfifo.....	create a FIFO special file	1-374
mknod.....	build a special file	1-376
more.....	display files on a page-by-page basis	1-378
mv.....	rename and move files and directories	1-386

Volume 2

Chapter 1. Commands and Utilities (continued)

nice.....	run a command at a different priority	1-391
nl.....	number lines	1-393
od.....	formatted file dump	1-397
pack.....	compress files by Huffman encoding	1-402
paste.....	horizontally concatenate lines	1-405
patch.....	change file using diff output	1-408
pathchk.....	check path names	1-414
pax.....	archiver for data interchange and file backup	1-416
pcat.....	display Huffman packed files on standard output	1-432
pr.....	display and format files	1-433
print.....	display arguments from the shell	1-437
printf.....	display a formatted string	1-439
ps.....	display process status	1-442
pwd.....	display working directory	1-451
r.....	edit and re-execute previous command	1-453
rcs.....	change RCS file attributes	1-456
rcsclean.....	clean up working files	1-466
rcsdiff.....	compare RCS revisions	1-470
rcsmerge.....	merge RCS revisions	1-474
read.....	input a line to the shell	1-478
readonly.....	mark variable as readonly	1-481
renice.....	set priorities of running processes	1-483
return.....	return from shell function or . (dot) script	1-486
rev.....	reverse character order of input lines	1-487
rlog.....	display info about RCS files	1-488
rm.....	remove files	1-494

rmdir.....	remove directory	1-497
sccs2rcs.....	SCCS to RCS conversion utility	1-499
sed.....	stream editor (non-interactive)	1-506
set.....	set shell flags and positional parameters	1-515
sh.....	POSIX-compliant (Korn) shell and command interpreter	1-519
rsh.....	POSIX-compliant (Korn) shell and command interpreter	1-519
shift.....	shift positional parameters	1-556
sleep.....	suspend execution for a specified time	1-557
sort.....	sort/merge utility	1-559
split.....	split a file into manageable pieces	1-567
strings.....	display printable strings in binary files	1-570
strip.....	remove debug information from executable files	1-572
stty.....	set or display terminal options	1-575
sum.....	compute checksum and block count for file	1-582
tabs.....	set terminal tab stops	1-584
tail.....	display last lines of file	1-588
tar.....	USTAR-compatible tape archiver to copy and back up files	1-590
tee.....	clone output stream	1-596
test.....	test for condition	1-598
tobyte.....	convert MPE record files to byte stream files	1-603
touch.....	change file modification date	1-605
tput.....	change terminal characteristics	1-608
tr.....	translation filter	1-611
trap.....	intercept abnormal conditions and interrupts	1-615
true.....	do nothing, successfully	1-617
tty.....	display terminal name	1-618
type.....	tell how shell interprets name	1-620
typeset.....	assign attributes and values to variables	1-621
umask.....	get or set the file mode creation mask	1-624
unalias.....	remove alias definitions	1-626
uname.....	display system name information	1-627
uncompress.....	Undo Lempel-Ziv compression of a file	1-629
unexpand.....	compress spaces into tabs	1-632
uniq.....	display unique lines of sorted file	1-634
unpack.....	decode Huffman packed files	1-637
unset.....	remove shell variable or function	1-639
uudecode.....	decode transmitted binary file	1-641
uuencode.....	encode a file for safe transmission	1-644
vi.....	display-oriented interactive text editor	1-646
wait.....	wait for process to complete	1-691
wall.....	write to all logged in users	1-692
wc.....	count of newlines, words, bytes, and characters	1-694
whence.....	tell how shell interprets command name	1-696
which.....	display path name for executable command	1-698
who.....	display information about current users	1-700

write	write to another user	1-702
xargs.....	construct and execute command lines	1-705
yacc	parser generator language	1-710
zcat	uncompress and display data	1-715

Chapter 2. File Formats

cpio.....	format of cpio archives	2-2
magic	format of the <code>/etc/magic</code> file	2-4
pax	format of pax archives	2-6
tags	format of tags file	2-7
tar	format of tar archives	2-8

Chapter 3. Miscellaneous Information

ascii	table of ASCII collating sequence	3-2
environ	standard environment variables	3-3
iconv	Available code sets for conversion	3-5
lex	functions used with lex	3-6
rcsedit	RCS editing facilities for descriptions and log messages	3-9
rcserror.....	RCS error messages	3-11
rcsfile	format of RCS file	3-24
references	related books and articles	3-27
regerror	regular expression error messages	3-29
regexp	syntax of regular expression patterns	3-32
shedit	interactive command and history editing in the shell	3-38
syserror.....	system error messages	3-46
timezone	setting local time zone	3-51
tmail	mail delivery program	3-54

Appendix A. MPE/iX Implementation Considerations

Permuted Index

Index

Tables

1-1	Escape Sequences in awk Literal Strings	1-20
1-2	awk Order of Operations	1-23
1-3	bc Operators	1-49
1-4	Table of RCS Keywords	1-113
1-5	Integer Values of Signals	1-280
1-6	Internal Table Sizes	1-287
1-7	Reference Notations	1-308
1-8	n1 Logical Page Delimiter Characters	1-393
1-9	wchan Abbreviations	1-448
1-10	Word Expansion	1-527
1-11	Shell Operators	1-534
2-1	ASCII Header Format for cpio File	2-2
2-2	Binary Header Format for cpio File	2-3
2-3	tar Header Block (TAR Format)	2-8
2-4	tar Header Block (USTAR Format)	2-9
2-5	Type Flag Values for USTAR Format Files	2-10
3-1	Standard 7-bit ASCII Representation	3-2
3-2	Available Code Sets	3-5
3-3	Regular Expression Features	3-36
A-1	Mapping Between POSIX.1 and ACD File Access Permissions	A-4
A-2	Mapping Between POSIX.1 and ACD Directory Access Permissions	A-4
A-3	Mapping Between POSIX.1 and ACD User Classes	A-5
A-4	MPE/iX Implementations of POSIX File Types	A-7
A-5	Signals Required by POSIX.1	A-15

Figures

3-1	An RCS Revision Tree	3-26
-----	----------------------------	------

Commands and Utilities

This chapter contains manual pages for all commands and utilities included in MPE/iX Shell and Utilities, arranged in alphabetical order (except for **intro(1)** which appears first). For details on how to read the manual page, see the **intro(1)** man page.

NAME

intro — introduction to man pages

DESCRIPTION

A description of an individual topic (for example, a command) is loosely called the *manual page* for that topic, even if it is actually several pages long. This is often abbreviated to *man page*, as in: “Read the man page for **ls**.” This man page describes the parts of a man page with examples taken from real MPE/iX Shell and Utilities man pages. Any of these parts may be omitted if they are irrelevant to the software being described.

When we refer to documentation in the manual, we usually give the topic followed by the chapter that contains the topic. For example, if we say “See **ls**(1)”, we mean that you should look up **ls** in *Chapter 1* of the *Reference Manual*. Since each chapter is in alphabetical order, it is quite easy to find anything in this way.

Name

The *NAME* section provides the name of the command and a brief functional description.

Synopsis

In the man page for a command, the *SYNOPSIS* section provides a quick summary of the command’s *format*. For example, here is the synopsis of the **ls** command.

```
ls [-abcCdfFgilmnopqrRstuxl] [pathname ...]
```

The synopsis takes the form of a command line as you might type it into the system; it shows what you can type in and the order in which you should do it. The parts that are enclosed in square brackets are *optional*; you may omit them if you choose. Parts that are not enclosed in square brackets *must* be present for the command to be correct.

The synopsis begins with the name of the command itself. In MPE/iX Shell and Utilities documentation, command names are always written in **bold Courier** font.

After the command name comes a list of options, if there are any. A typical MPE/iX Shell and Utilities command option consists of a dash (–) followed by a single character, usually an uppercase or lowercase letter. For example, you might have **–l** or **–t**. If you are going to specify several options for the same command, you can put all the option characters after the same dash; for example,

```
ls -l -t
ls -lt
ls -t -l
ls -tl
```

are all equivalent.

The synopsis line shows options in **bold Courier** font. Note that the case of letters is important; for example, in the synopsis of **ls**, **-f** and **-F** are *different* options, with different effects.

In the description of **ls**, all options are shown in one long string after the single dash. Another common option form is

```
-x value
```

where **-x** is a dash followed by a character, and *value* provides extra information for using that option. For example, the **sort** command takes unsorted input and sorts it; here's the command's synopsis:

```
sort [-cmu] [-o outfile] [-tchar] [-yn] [-zn] [-bdfiMnr]  
[-k startpos[,endpos]] ... [file ...]
```

In this example, we have the option

```
-o outfile
```

This option tells the **sort** command where to save its sorted output. The form of the option is **-o**, followed by a space, followed by *outfile*. In a command synopsis, anything appearing in *italics* is a *placeholder* for information that you are expected to supply. Sometime after the synopsis, the man page will explain what kind of information is expected in place of the placeholder. In our **sort** example, *outfile* stands for the name of a file where you want **sort** to store its output. For example, if you wanted to store the output in the file `sorted.dat`, you would specify

```
sort -o sorted.dat
```

(followed by the rest of the command).

You will notice that the synopsis for **sort** also contains an option of the form

```
-tchar
```

This is similar to the option form we were just discussing, except that there is no space between the **-t** and *char*. As before, *char* in italics is a placeholder; in this case, it stands for any single character. If you want to use the **-t** option for **sort**, you just type **-t** followed immediately by another character, as in

```
sort -t:
```

In this case, we use a colon (:) in the position of the placeholder *x*.

The end of the **sort** synopsis is

```
[file . . .]
```

This means a list of one or more file names; the ellipsis (. . .) stands for repetitions of whatever immediately precedes it. Since there are square brackets around the previous list, the list can be omitted if you like.

The synopsis of **ls** ended in

```
[pathname . . .]
```

As you might guess, this means that an **ls** command may end with an optional list of one or more path names. (What's the difference between this and our **sort** example? A *pathname* may be the name of either a file or a directory; a *file* is always the name of a file.)

The order of items on the command line is important. When you type in a command line, you should specify the parts of the command line in the order they appear in the command synopsis. The exceptions to this are options marked with a - ; they do not have to be given in the exact order shown in the synopsis. However, all the - options must appear in the correct *area* of the command line. For example, you can specify

```
ls -l -t myfiles
ls -t -l myfiles
```

but you won't get correct results if you specify

```
ls myfiles -l -t          ***incorrect***
ls -l myfiles -t          ***incorrect***
```

and so on. If you type the last command, for example, **ls** interprets **-t** as the path name of a file/directory and the command will try to list the characteristics of that item.

As a special notation, most MPE/iX Shell and Utilities commands let you specify **--** to separate the options from the non-option arguments; **--** means: "There are no more options." Thus if you really have a directory named **-t**, you could specify

```
ls -- -t
```

to list the contents of that directory.

Description

The *DESCRIPTION* section describes what the command does and how each of the options work. For particularly complex software, this section may be divided into a large number of subsections, each dealing with a particular aspect of the command.

The *DESCRIPTION* section often mentions the *standard input* and the *standard output*. The standard input is usually the terminal keyboard; the standard output is usually the display screen. The process of *redirection* can change this. Redirection is explained in the glossary of the *User's Guide*, and in other parts of the MPE/iX Shell and Utilities documentation.

When a utility reads data from the standard input, it accepts the lines that you type on the keyboard as if they were lines from files. To end a line of input, press ENTER. To indicate the end of all the input (that is, the end of file), enter the end-of-file character. On MPE/iX, use the string :EOD to indicate the end of the file.

Inside the *DESCRIPTION* section, the names of files and directories are written in normal Courier font. The names of environment variables are written in *italic Courier* font.

Examples

The *EXAMPLES* section is present in many man pages, giving examples of how the software can be used. We try to give a mix of simple examples that show how the commands work on an elementary level, and more complex examples that show how the commands can perform complicated tasks.

Environment Variables

The *ENVIRONMENT VARIABLES* section lists the environment variables that affect the command, if any, and describes the purposes that those variables serve. For example, the **ls** man page lists two environment variables *COLUMNS* and *TZ* and informs you that *COLUMNS* is the terminal width and that *TZ* contains information about the local time zone.

Files

The *FILES* section lists the supplementary files that the command refers to, if any. By supplementary files, we mean files that are not specified on the command line. Such files usually provide information that the command needs; the command accesses these files during its operation. If the files cannot be found, the command prints a message to this effect.

Files documented in this section may be temporary files, output files, databases, configuration files, and so on.

Diagnostics

The *DIAGNOSTICS* section contains information about the exit status returned by the command. You can test this status to determine the result of the operation that the command was asked to perform. The *Messages* subsection presents the error messages that the software may display, along with a description of what caused the message and a possible action you can take to avoid getting that message. Occasionally, one man page will refer you to another for more information on an error message. Three common man pages that you will be referred to

are the **regerror**(3) man page which describes all the errors that occur while processing regular expressions, the **rcserror**(3) man page which describes errors that are common to most RCS utilities and the **syserror**(3) man page which describes system errors that are produced by the operating system.

Portability

The *PORTABILITY* section includes two types of information:

- Availability of a version of the command on existing UNIX[®] systems (System V, BSD).
- Compatibility with industry standards (for example, the POSIX.2 Standard or the *X/OPEN* Portability Guide, Issue 4).

Limits

The *LIMITS* section lists any limits on the operation of the software. For example, the **dc**(1) command is intended to work like a desk calculator that can handle numbers of any size. However, at the time of this writing, it actually has a limit of 1000 digits when typing in a single number. Limits of this sort are inevitable when writing software, but when designing MPE/iX Shell and Utilities, we did our best to set the limits high enough that they will not get in the way of our users.

Some limits are implicit rather than explicit, and may be lower than the explicitly stated limit.

Warning

The *WARNING* section contains important advice for users. In MPE/iX Shell and Utilities documentation, the *WARNING* section is often aimed at those who are familiar with UNIX systems. Since MPE/iX Shell and Utilities complies, first and foremost, to the POSIX standards, and resides on a proprietary platform, its behavior may not precisely match the corresponding UNIX commands. The *WARNING* section may point out discrepancies in behavior that may catch experienced POSIX or UNIX users by surprise.

Notes

The *NOTES* section gives additional notes for those using the software. The purpose of the *NOTES* section is similar to that of the *WARNING* section — to provide important information that the reader should not overlook; however, *NOTES* usually deal with issues that are more benign than *WARNINGS*.

MPE/iX Notes

The *MPE/iX NOTES* section lists information that applies only to the MPE/iX version of this software. Since the current release of MPE/iX is not fully POSIX.1 compliant, it places a number of limitations on the MPE/iX Shell and Utilities software. The limitations on a given utility are listed in the *MPE/iX NOTES* section of that utility's man page.

See Also

The *SEE ALSO* section refers to other man pages that may contain information relevant to the man page you have just read. For example, consider the **compress** command; this command helps you *shrink* data files into a compact form to save storage space. Its *SEE ALSO* section refers you to **uncompress(1)**, the command that restores shrunken data files to their original state.

Permuted Index

The permuted index is not really part of the manual pages, but rather a guide to the manual pages. It precedes the Index at the end of each volume of the *MPE/iX Shell and Utilities Reference Manual*. The permuted index helps you explore MPE/iX Shell and Utilities and what its commands can do for you.

Each line in the permuted index is taken from the title of a manual page. The words of the title are shifted to get entries for the index. For example, the title of the **comm** manual page is

```
comm - compare sorted files and show differences
```

This produces the following permuted index entries:

```

          show differences/   comm - compare sorted files and
              differences/   compare sorted files and show
compare sorted files and show differences
              compare sorted files and show differences
          compare sorted files and show differences
              compare        sorted files and show differences

```

Notice that there is a gap in the middle of each line. The permuted index is sorted by the word that comes after this gap. Also notice that the end of the original title line is marked with a slash (/) when it doesn't appear at the end of the index line.

The permuted index lets you look up commands according to key words in their title lines. For example, suppose you want to compare two files and want to know which commands do this. Looking up the word *compare* in the permuted index, we find several entries:

```

          differences/   compare sorted files and show
                          compare three text files
minimal differences/   compare two text files and show
          binary        comparison of two files

```

Each of these index entries is associated with a manual page (given at the end of the index entry line). By looking at the title lines, you should be able to determine which command does what you want and go directly to the appropriate manual page.

NAME

alias — display or create command aliases

SYNOPSIS

```
alias -tx [name[=value] ...]
```

DESCRIPTION

When the first word of a shell command line is not a shell keyword, the shell checks for the word in the list of currently defined aliases. If it finds a match, the shell replaces the alias with its associated string value. The result is a new command line that might begin with a shell function name, a built-in command, an external command or another alias.

When the shell performs alias substitution, it checks to see if *value* ends with a blank. If so, the shell also checks the next word of the command line for aliases. The shell then checks the new command line for aliases and expands them, following these same rules. or recursion occurs in the expansion of aliases.

Calling **alias** without parameters displays all the currently defined aliases and their associated values. Values appear with appropriate quoting so that they are suitable for re-input to the shell.

Calling **alias** with parameters in the form of

```
name=value
```

creates an alias for each *name* with the given string *value*.

If you are defining an alias where *value* contains a backslash character, you must precede it with another backslash since when the shell performs the expansion, it interprets a backslash as the escape character. If you use double quotes to enclose *value*, you must precede each component of a double backslash with an additional backslash since the shell escapes characters both when assigning the alias and again when expanding it

To avoid using four backslashes to represent a single backslash, use apostrophes rather than double quotes to enclose *value*, since the shell does not escape characters enclosed in apostrophes during assignment. As a result, the shell only escapes characters within apostrophes when expanding the alias.

Calling **alias** with *name* without any *value* assignment, displays *name* and its associated *value* with appropriate quoting.

Options

alias accepts the following options:

- t** makes each *name* on the command line a *tracked* alias. Each tracked alias resolves to its full path name; thus the shell avoids searching the *PATH* directories whenever you invoke the command. The shell assigns the full path name of a tracked alias the first time that you invoke it. It reassigns a path name the first time you use the alias after changing the variable *PATH* or running the shell command **cd**. When you issue the command

```
set -h
```

each subsequent command you use in the shell automatically becomes a tracked alias. Invoking **alias** with the **-t** option, but without any specified *names*, displays all currently defined tracked aliases with appropriate quoting.

- x** marks each alias *name* on the command line for export. If you specify **-x** without any *names*, **alias** displays all exported aliases. Only exported aliases are passed to a shell that runs a shell script.

Built-in Aliases

There are several aliases built into the shell:

```
alias functions="typeset -f"  
alias hash="alias -t"  
alias history="fc -l"  
alias integer="typeset -i"  
alias r="fc -s"
```

On systems supporting job control:

```
alias stop="kill -STOP"  
alias suspend="stop \${\$}"
```

You can change or remove any of these aliases. See the relevant manual pages for details.

EXAMPLES

The command:

```
alias ls="ls -C"
```

defines **ls** as an alias. From this point onward, when you issue an **ls** command, it produces multi-column output by default.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure because an alias could not be set.
- 2 Failure because of an invalid command line option.

If you define **alias** to determine the values of a set of names, the exit status is the number of those names which are not currently defined as aliases.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh(1)** man page for a complete list of error messages that you may receive when using it.

PORTABILITY

KornShell. POSIX.2. *x/OPEN* Portability Guide 4.0.

On UNIX systems, **alias** is a built-in command of the KornShell, but not of the Bourne Shell.

The **-t** and **-x** options are extensions to the POSIX standard.

NOTE

This command is built into the shell.

Since exported aliases are only available in the current shell environment and to the child processes of this environment, they will be unavailable to any new shell environment which are started. To make an alias available to all shell environments, define it as a non-exported alias in the MPE/iX Shell *ENV* file, which is executed whenever a new shell is run.

SEE ALSO

cd(1), **fc(1)**, **functions(1)**, **hash(1)**, **history(1)**, **integer(1)**, **let(1)**, **pwd(1)**, **r(1)**, **set(1)**, **sh(1)**, **typeset(1)**, **unalias(1)**

NAME

ar — create and maintain library archives

SYNOPSIS

```
ar -d [-v] archive member ...
ar -r [-cuvt] archive member ...
ar -t [-v] archive [member ...]
ar -x [-v] archive [member ...]
```

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

ar maintains archive libraries. You use **ar** to

- create a new library
- add members to an existing library
- delete members from a library
- extract members from a library
- print a table of contents for a library

A library member must be an object file, suitable for use by a link editor. **ar** creates and maintains an external symbol index, to allow the library to be used for link editing.

Member names in an archive are only the final component of any file name. When opening a *member* as given on the command line, the full path name given is used. When storing the member name in the library, or comparing a member name, only the final component is used.

Options

The synopsis shows the four main functions of **ar**, defined as follows:

- d deletes each named *member* from the archive and regenerates the symbol table.
- r replaces or creates a new archive. If *archive* does not exist, **ar** creates it and prints a message. This option adds each specified *member* file as a module to the archive and regenerates the symbol table.
- t displays a table of contents that lists *members* or every member if no *member* is specified. **ar** prints a diagnostic for each member that it doesn't find. By default, **ar** prints the member name for all selected members. With the verbose (-v) option, **ar** prints more information for all selected members.
- x extracts each specified *member* from the archive and copies them (or all members if none specified) to a file of the full *member* name given.

The following options modify the behavior of the main functions:

- c** suppresses the normally printed message when a new *archive* file is created. You can only use this in conjunction with the **-r** option.
- u** when used with **-r**, replaces the archive member only if the *member* file's modification time is more recent than the archive member time.
- v** prints the command letter and the member name affected before performing each operation. With **-t**, **ar** prints more information about archive members using a format similar to **ls -l**.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** *archive* is not a valid library (file code should be NMRL)
Cause: The file code on the specified file was not NMRL
Action: Make sure that you have specified the correct file, and that it is a valid native-mode relocatable library.
- Message:** can only specify one of 'drtx'
Cause: You specified more than one of the **-d**, **-r**, **-t**, and **-x** options on the command line. You can specify only one of these options at a time.
Action: Consult the *DESCRIPTION* section and correct the command line.
- Message:** can only specify u or c with r
Cause: You specified the **u** or **c** option modifier without specifying the **-r** option. These option modifiers can only be used with the **-r** option.
Action: Consult the *DESCRIPTION* section and correct the command line.
- Message:** can't open library '*archive*' for modification
Cause: The program was unable to open the file with write access.
Action: Make sure that you have the necessary permissions to write this file.
- Message:** can't open library '*archive*' for reading
Cause: The program was unable to open the file with read access.
Action: Make sure that you have the necessary permissions to read this file.

-
- Message:** cannot access file '*filename*' - file ignored
Cause: The program was unable to access the specified file for reading, so it was not added to the library.
Action: Acquire read access to the file and run the **ar** command again.
- Message:** *filename* is not a valid object module (file code should be NMOBJ)
Cause: The file code on the specified file was not NMOBJ
Action: Make sure that you have specified the correct file, and that it is a valid native-mode object file.
- Message:** library *archive* doesn't exist
Cause: The specified library file could not be found.
Action: Make sure that *archive* exists, is spelled correctly, and, if necessary, the correct path is given.
- Message:** must specify one of 'drtx'
Cause: You failed to specify one of the **-d**, **-r**, **-t**, or **-x** options on the command line. You must specify one of these options.
Action: Consult the *DESCRIPTION* section and correct the command line.
- Message:** name of archive library must be specified
Cause: You failed to specify the name of a library archive to use as the first operand on the command line.
Action: Consult the *DESCRIPTION* section and correct the command line.
- Message:** unknown option *-option*
Cause: You specified an option that is not valid for **ar**.
Action: Check the *DESCRIPTION* section for a list of valid **ar** options.

For specific link editor error messages, refer to Appendix A of the *HP Link Editor/iX Reference Manual* (Part Number 32650-90030).

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

MPE/iX NOTES

The **ar** command is currently implemented on MPE/iX as a script-driven front-end to the MPE/iX link editor utility.

Refer to the *HP Link Editor/iX Reference Manual* (Part Number 32650-90030) for details on link editor operation and diagnostic messages.

If the environment variable *ECHO* is defined, the **ar** utility displays the commands that are passed to the MPE/iX CI for execution. For example

```
$ ECHO=1 ar -r mylib.a func1.o func2.o
```

displays each of the commands submitted to the MPE/iX CI.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

c89(1) **make(1)**

NAME

asa — interpret ASA/FORTRAN carriage control

SYNOPSIS

asa [*file ...*]

DESCRIPTION

Historically, print-outs created by ASA/FORTRAN programs use the first character of each line to control the spacing between that line and the previous one. For example, if the first character is a blank, the rest of that line immediately follows the previous line; if it is a 1, that line should begin on a new page, and so on.

The purpose of **asa** is to read input in this format and write it out in a normal text format, using newlines, formfeeds, and carriage returns to achieve the same effects as the ASA/FORTRAN carriage control characters.

If you specify files on the command line, **asa** reads input from these files; otherwise, it reads the standard input. **asa** writes output to the standard output.

This utility does not copy newline characters in the input to the output. Instead, it uses the first character of each line to determine how to print the rest of the line. **asa** interprets the first character as follows, where *line* is the rest of the line after the first character.

Blank outputs a single newline character before printing *line*.

0 outputs two newline characters before printing *line*.

1 outputs a formfeed (start a new page) before printing *line*.

+

 outputs a carriage return before printing *line*, (overprinting the previous *line*).

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - write error on the standard output
 - the inability to open the input file
- 2 Unknown option specified on the command line.

Messages

Message: asa: write error on standard output: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

Message: asa: input file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **asa**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **asa** options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. UNIX System V.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

NAME

awk — data transformation, report generation language

SYNOPSIS

```
awk [-F ere] [-f prog] [-v var=value ...] [program] [var=value ...] [file ...]
```

DESCRIPTION

awk is a file-processing language which is well-suited to data manipulation and retrieval of information from text files. This reference page provides a full technical description of **awk**. If you are unfamiliar with the language, you may find it helpful to read the **awk Tutorial** in the *User's Guide* before reading the following material.

An **awk** program consists of any number of user-defined functions and *rules* of the form:

```
pattern { action }
```

There are two ways to specify the **awk** program:

- (a) Directly on the command line. In this case, the *program* is a single command line argument, usually enclosed in apostrophes (') to prevent the shell from attempting to expand it.
- (b) By using the **-f prog** option.

You can only specify *program* directly on the command line if you do not use any **-f prog** arguments.

When you specify *files* on the command line, those files provide the input data for **awk** to manipulate. If you specify no such files or you specify **-** as a file, **awk** reads data from the standard input.

You can initialize variables on the command line using

```
var=value
```

You can intersperse such initializations with the names of input files on the command line. **awk** processes initializations and input files in the order they appear on the command line. For example, the command

```
awk -f progfile a=1 f1 f2 a=2 f3
```

sets *a* to 1 before reading input from *f1* and sets *a* to 2 before reading input from *f3*.

Variable initializations that appear before the first *file* on the command line are performed immediately after the **BEGIN** action. Initializations appearing after the last *file* are performed immediately before the **END** action. For more information on **BEGIN** and **END**, see *Patterns*.

The **-v** option lets you assign a value to a variable before the **awk** program begins running (that is, before the BEGIN action). For example, in

```
awk -v v1=10 -f prog datafile
```

awk assigns the variable `v1` its value before the BEGIN action of the program (but after default assignments made to built-in variables like FS, and OFMT; these built-in variables have special meaning to **awk**, as described in later sections).

awk divides input into *records*. By default, newline characters separate records; however, you may specify a different record separator if you want.

One at a time, and in order, **awk** compares each input record with the pattern of every rule in the program. When a pattern matches, **awk** performs the action part of the rule on that input record. Patterns and actions often refer to separate *fields* within a record. By default, white space (usually blanks, newlines, or horizontal tab characters) separates fields; however, you can specify a different field separator string using the **-F** *ere* option (see *Input*).

You can omit the *pattern* or *action* part of an **awk** rule (but not both). If you omit *pattern*, **awk** performs the *action* on every input record (that is, every record matches). If you omit *action*, **awk** writes every record matching the *pattern* to the standard output.

awk considers everything after a # in a program line to be a comment. For example:

```
# This is a comment
```

To continue program lines on the next line, add a backslash (\) to the end of the line. Statement lines ending with a comma (,), double or-bars (| |), or double ampersands (&&) continue automatically on the next line.

Options

awk accepts the following options:

-F *ere* specifies an extended regular expression to use as the field separator.

-f *prog*

runs the **awk** program contained in the file *prog*. When more than one **-f** option appears on the command line, the resulting program is a concatenation of all programs you specify.

-v *var=value*

assigns *value* to *var* before running the program. You can specify this option a number of times.

Variables and Expressions

There are three types of *variables* in **awk**: *identifiers*, *fields*, and *array elements*.

An *identifier* is a sequence of letters, digits, and underscores beginning with a letter or an underscore.

For a description of *fields*, see the *Input* subsection.

Arrays are associative collections of values called the *elements* of the array. Constructs of the form,

identifier [*subscript*]

where *subscript* has the form *expr* or *expr,expr,...*, reference array elements. Each such *expr* can have any string value. For multiple *expr* subscripts, **awk** concatenates the string values of all *exprs* with a separate character SUBSEP between each. The initial value of SUBSEP is set to \034 (ASCII field separator).

We sometimes refer to fields and identifiers as *scalar variables* to distinguish them from arrays.

You do not declare **awk** variables and you do not need to initialize them. The value of an uninitialized variable is the empty string in a string context and the number 0 in a numeric context.

Expressions consist of constants, variables, functions, regular expressions and *subscript in array* conditions (described later) combined with operators. Each variable and expression has a string value and a corresponding numeric value; **awk** uses the value appropriate to the context.

When converting a numeric value to its corresponding string value, **awk** performs the equivalent of a call to the `printf` function (see *Built-in String Functions*) where the one and only *expr* argument is the numeric value and the *fmt* argument is either `%d` (if the numeric value is an integer) or the value of the variable `CONVFMT` (if the numeric value is not an integer). The default value of `CONVFMT` is `%.6g`. If you use a string in a numeric context, and **awk** cannot interpret the contents of the string as a number, it treats the value of the string as zero.

Numeric constants are sequences of decimal digits.

String constants are quoted, as in "a literal string". Literal strings can contain the escape sequences shown in Table 1-1, *Escape Sequences in awk Literal Strings*.

Escape	Character
<code>\a</code>	audible bell
<code>\b</code>	backspace
<code>\f</code>	formfeed
<code>\n</code>	newline
<code>\r</code>	carriage return
<code>\t</code>	horizontal tab
<code>\v</code>	vertical tab
<code>\ooo</code>	octal value <i>ooo</i>
<code>\xdd</code>	hexadecimal value <i>dd</i>
<code>\/</code>	slash
<code>\"</code>	quote
<code>\c</code>	any other character <i>c</i>

Table 1-1: Escape Sequences in **awk** Literal Strings

awk supports full regular expressions (see **regexp(3)**). When **awk** reads a program, it compiles characters enclosed in slash characters (/) as regular expressions. In addition, when literal strings and variables appear on the right side of a `~` or `!~` operator, or as certain arguments to built-in matching and substitution functions, **awk** interprets them as dynamic regular expressions.

Note: When you use literal strings as regular expressions, you need extra backslashes to escape regular expression metacharacters, since the backslash is also the literal string escape character. For example the regular expression,

```
/e\.g\./
```

when written as a string is:

```
"e\\.g\\.."
```

awk defines the *subscript in array* condition as:

```
index in array
```

where *index* looks like *expr* or (*expr*,...,*expr*). This condition evaluates to 1 if the string value of *index* is a subscript of *array*, and to 0 otherwise. This is a way to determine if an array element exists. When the element does not exist, this condition does not create it.

Symbol Table

You can access the symbol table through the built-in array `SYMTAB`.

```
SYMTAB[expr]
```

is equivalent to the variable named by the evaluation of *expr*. For example,

```
SYMTAB["var"]
```

is a synonym for the variable `var`.

Environment

An **awk** program can determine its initial environment by examining the `ENVIRON` array. If the environment consists of entries of the form:

```
name=value
```

then

```
ENVIRON[name]
```

has string value

```
"value"
```

For example, the following program is equivalent to the default output of `env(1)`:

```
BEGIN {
  for (i in ENVIRON)
    printf("%s=%s\n", i, ENVIRON[i])
  exit
}
```

Operators

awk follows the usual precedence order of arithmetic operations, unless overridden with parentheses; a table giving the order of operations appears later in this section.

The unary operators are `+`, `-`, `++`, and `--`, where you can use the `++` and `--` operators as either postfix or prefix operators, as in C. The binary arithmetic operators are `+`, `-`, `*`, `/`, `%`, and `^`.

The conditional operator

```
expr ? expr1 : expr2
```

evaluates to the *expr1* if the value of *expr* is non-zero, and to *expr2* otherwise.

If two expressions are not separated by an operator, **awk** concatenates their string values.

The operator `~` yields 1 (true) if the regular expression on the right side matches the string on the left side. The operator `!~` yields 1 when the right side has no match on the left. To illustrate:

```
$2 ~ /[0-9]/
```

selects any line where the second field contains at least one digit. **awk** interprets any string or variable on the right side of `~` or `!~` as a dynamic regular expression.

The relational operators are `<`, `<=`, `>`, `>=`, `==`, and `!=`. When both operands in a comparison are numeric, **awk** compares their values numerically; otherwise, it compares them as strings. An operand is numeric if it is an integer or floating point number, if it is a field or ARGV element that looks like a number, or if it is a variable created by a command line assignment that looks like a number.

The Boolean operators are `||` (or), `&&` (and), and `!` (not). Short Circuit Evaluation is used when evaluating expressions. With an `&&` expression, if the first operator is false, the entire expression is false and it is not necessary to evaluate the second operator. With an `||` expression, a similar situation exists if the first operator is true.

You can assign values to a variable with

```
var = expr
```

If *op* is a binary arithmetic operator,

```
var op= expr
```

is equivalent to

```
var = var op expr
```

except that *var* is evaluated only once.

See Table 1-2, *awk Order of Operations* for the precedence rules of the operators.

Order of Operations	
(A)	grouping
$\$i$ V[a]	field, array element
V++ V-- ++V --V	increment, decrement
A^B	exponentiation
+A -A !A	unary plus, unary minus, logical NOT
A*B A/B A%B	multiplication, division, remainder
A+B A-B	addition, subtraction
A B	string concatenation
A<B A>B A<=B A>=B A!=B A==B	comparisons
A~B A!~B	regular expression matching
A in V	array membership
A && B	logical AND
A B	logical OR
A ? B : C	conditional expression
V=B V+=B V-=B V*=B V/=B V%=B V^=B	assignment
A, B and C are any expression. <i>i</i> is any expression yielding an integer. V is any variable.	

Table 1-2: **awk** Order of Operations

Command Line Arguments

awk sets the built-in variable ARGV to the number of command line arguments. The built-in array ARGV has elements subscripted with digits from zero to ARGV-1, giving command line arguments in the order they appeared on the command line.

The ARGV count and the ARGV vector do not include command line options (beginning with -) or the program file (following -f). They do include the name of the command itself, the names of input data files, and initialization statements of the form

var=value

awk actually creates ARGV and ARGV before doing anything else. It then walks through ARGV processing the arguments. If an element of ARGV is an empty string, **awk** skips it. If it contains an equals sign (=), **awk** interprets it as a variable assignment. If it is a minus sign (-), **awk** immediately reads input from the standard input until it encounters the end-of-file; otherwise, **awk** treats the argument as a file name and reads input from that file until it reaches end-of-file.

Note: **awk** runs the program by *walking through* ARGV in this way; thus if the program changes ARGV, **awk** can read different files and make different assignments.

Input

awk divides input into records. A *record separator character* separates each record from the next. The value of the built-in variable RS gives the current record separator character; by default, it begins as the newline (\n). If you assign a different character to RS, **awk** uses that as the record separator character from that point on.

awk divides records into fields. A *field separator string*, given by the value of the built-in variable FS, separates each field from the next. You can set a specific separator string by assigning a value to FS, or by specifying the **-F** *ere* option on the command line. You can assign a regular expression to FS. For example,

```
FS = "[, :$]"
```

says that commas, colons, or dollar signs can separate fields. As a special case, assigning FS a string containing only a blank character sets the field separator to white space. In this case, **awk** considers any sequence of contiguous space and/or tab characters a single field separator. This is the default for FS; however, if you assign FS a string containing any other character, that character designates the start of a new field. For example, if we set FS="\t" (the tab character),

```
texta \t textb \t \t \t textc
```

contains five fields, two of which only contain blanks. With the default setting, this record only contains three fields, since **awk** considers the sequence of multiple blanks and tabs a single separator.

The following list of built-in variables provides various pieces of information about input.

```
NF          number of fields in the current record
NR          number of records read so far
FILENAME    name of file containing current record
FNR        number of records read from current file
```


Field specifiers have the form $\$n$ where n runs from 1 through NF. Such a field specifier refers to the n th field of the current input record. $\$0$ (zero) refers to the entire current input record.

The `getline` function can read a value for a variable or $\$0$ from the current input, from a file, or from a pipe. The result of `getline` is an integer indicating whether the read operation was successful. A value of 1 indicates success; 0 indicates end-of-file encountered; and -1 indicates that an error occurred. Possible forms for `getline` are:

`getline`

reads next input record into $\$0$ and splits the record into fields. NF, NR, and FNR are set appropriately.

`getline var`

reads next input record into the variable *var*. **awk** does not split the record into fields (which means that the current $\$n$ values do not change), but sets NR and FNR appropriately.

`getline <expr`

interprets the string value of *expr* to be a file name. **awk** reads the next record from that file into $\$0$, splits it into fields, and sets NF appropriately. If the file is not open, **awk** opens it. The file remains open until you close it with a `close` function.

`getline var <expr`

interprets the string value of *expr* to be a file name, and reads the next record from that file into the variable *var*, but does not split it into fields.

expr | `getline`

interprets the string value of *expr* as a command line to be run. **awk** pipes output from this command into `getline`, and reads it into $\$0$ in a manner similar to `getline <expr`. See the *System Function* section for additional details.

expr | `getline var`

runs the string value of *expr* as a command and pipes the output of the command into `getline`. The result is similar to `getline var <expr`.

You can only have a limited number of files and pipes open at one time. You can close files and pipes during execution using the

`close(expr)`

function. The *expr* must be one that came before | or after < in `getline`, or after > or >> in `print` or `printf`. For a description of `print` and `printf`, see the *Output* section. If the function successfully closes the pipe, it returns zero. By closing files and pipes that you no longer need, you can use any number of files and pipes in the course of running an **awk** program.

Built-In Arithmetic Functions

`atan2(expr1, expr2)`

returns the arctangent of *expr1* / *expr2* in the range of $-\pi$ through π .

`exp(expr)`, `log(expr)`, `sqrt(expr)`

returns the exponential, natural logarithm, and square root of the numeric value of *expr*. If you omit (*expr*), these functions use \$0 instead.

`int(expr)`

returns the integer part of the numeric value of *expr*. If you omit (*expr*), the function returns the integer part of \$0.

`rand()`

returns a random floating-point number in the range 0 through 1.

`sin(expr)`, `cos(expr)`

returns the sine and cosine of the numeric value of *expr* (interpreted as an angle in radians).

`srand(expr)`

sets the seed of the `rand` function to the integer value of *expr*. If you omit (*expr*), **awk** uses the time of day as a default seed.

Built-In String Functions

`n = gsub(regexp, repl, string)`

works the same way as `sub`, except that `gsub` replaces all matching substrings (global substitution).

`pos = index(string, str)`

returns the position of the first occurrence of *str* in *string*. If `index` does not find *str* in *string*, it returns zero.

`len = length(expr)`

returns the number of characters in the string value of *expr*. If you omit (*expr*), the function uses \$0 instead. The parentheses around *expr* are optional.

`pos = match(string, regexp)`

searches *string* for the first substring matching the regular expression *regexp*, and returns an integer giving the position of this substring counting from one. If it finds no such substring, `match` returns zero. This function also sets the built-in variable `RSTART` to *pos* and the built-in variable `RLENGTH` to the length of the matched string. If it does not find a match, `match` sets `RSTART` to zero and `RLENGTH` to -1. You can enclose *regexp* in slashes or specify it as a string.

`n = ord(expr)`
 returns the integer value of first character in the string value of *expr*. This is useful in conjunction with `%c` in `sprintf`.

`n = split(string, array, regex)`
 splits the *string* into fields. *regex* is a regular expression giving the field separator string for the purposes of this operation. This function assigns the separate fields, in order, to the elements of *array*; subscripts for *array* begin at 1. **awk** discards all other elements of *array*. `split` returns the number of fields into which it divided *string* (which is also the maximum subscript for *array*). *regex* divides the record in the same way that the FS field separator string does. If you omit *regex* in the call to `split`, it uses the current value of FS.

`str = sprintf(fmt, expr, expr...)`
 formats the expression list *expr*, *expr*, ... using specifications from the string *fmt*, then returns the formatted string. The *fmt* string consists of conversion specifications which convert and add the next *expr* to the string, and ordinary characters which `sprintf` simply adds to the string. These conversion specifications are similar to those used by the ANSIC standard.

Conversion specifications have the form

`%[-][0][x][.y]c`

where

-	left justifies the field; default is right justification
0	leading zero prints numbers with leading zero
x	is the minimum field width
y	is the precision
c	is the conversion character

In a string, the precision is the maximum number of characters to be printed from the string; in a number, the precision is the number of digits to be printed to the right of the decimal point in a floating point value. If *x* or *y* is * (asterisk), the minimum field width or precision is the value of the next *expr* in the call to `sprintf`.

The conversion character *c* is one of following:

d	decimal integer
i	decimal integer
o	unsigned octal integer
x, X	unsigned hexadecimal integer
u	unsigned decimal integer
f, F	floating point
e, E	floating point (scientific notation)

g, G the shorter of e and f (suppresses non-significant zeros)
 c single character of an integer value; first character of string
 s string

The lowercase x prints alphabetic hex digits in lowercase while the uppercase X prints alphabetic hex digits in uppercase. The other upper/lowercase pairs work similarly.

n = sub(*regexp*, *repl*, *string*)
 searches *string* for the first substring matching the extended regular expression *regexp*, and replaces the substring with the string *repl*. **awk** replaces any ampersand (&) in *repl* with the substring of *string* which matches *regexp*. You can suppress this special behavior by preceding the ampersand with a backslash. If you omit *string*, sub uses the current record instead. sub returns the number of substrings replaced (which is one if it found a match, and zero otherwise).

str = substr(*string*, *offset*, *len*)
 returns the substring of *string* that begins in position *offset* and is at most *len* characters long. The first character of the string has an *offset* equal to one. If you omit *len*, substr returns the rest of *string*.

str = tolower(*expr*)
 converts all letters in the string value of *expr* into lowercase, and returns the result. If you omit *expr*, tolower uses \$0 instead.

str = toupper(*expr*)
 converts all letters in the string value of *expr* into uppercase, and returns the result. If you omit *expr*, toupper uses \$0 instead.

System Function

status = system(*expr*)
 runs the string value of *expr* as a command. For example,

```
system("tail " $1)
```

calls the **tail** command, using the string value of \$1 as the file that **tail** examines. The MPE/iX Shell runs the command as discussed in the *PORTABILITY* section, and the exit status returned depends on that command interpreter.

User-Defined Functions

You can define your own functions using the form

```
function name(parameter-list) {
    statements
}
```

A function definition can appear in the place of a *pattern* {*action*} rule. The *parameter-list* contains any number of normal (scalar) and array variables separated by commas. When you call a function, **awk** passes scalar arguments by value, and array arguments by reference. The names specified in the *parameter-list* are local to the function; all other names used in the function are global. You can define local variables by adding them to the end of the parameter list as long as no call to the function uses these extra parameters.

A function returns to its caller either when it performs the final statement in the function, or when it reaches an explicit `return` statement. The return value, if any, is specified in the `return` statement (see the *Actions* section).

Patterns

A *pattern* is a regular expression, a special pattern, a pattern range, or any arithmetic expression.

BEGIN is a special pattern used to label actions that **awk** performs before reading any input records. END is a special pattern used to label actions that **awk** performs after reading all input records.

You can give a pattern range as

```
pattern1 , pattern2
```

This matches all lines from one that matches *pattern1* to one that matches *pattern2*, inclusive.

If you omit a pattern, or if the numeric value of the pattern is non-zero (true), **awk** performs the resulting action for the line.

Actions

An *action* is a series of statements terminated by semicolons, newlines, or closing braces. A *condition* is any expression; **awk** considers a non-zero value true, and a zero value false. A *statement* is one of the following or any series of *statements* enclosed in braces.

```
# expression statement, e.g. assignment  
expression
```

```
# if statement  
if (condition)  
    statement  
[else  
    statement ]
```

```
# while loop  
while (condition)  
    statement
```

```
# do-while loop
do
    statement
while (condition)

# for loop
for (expression1 ; condition ; expression2)
    statement
```

The for statement is equivalent to:

```
expression1
while (condition) {
    statement
    expression2
}
```

The for statement can also have the form

```
for (i in array)
    statement
```

awk performs the *statement* once for each element in *array*; on each repetition, the variable *i* contains the name of a subscript of *array*, running through all the subscripts in an *arbitrary* order. If *array* is multi-dimensional (has multiple subscripts), *i* is expressed as a single string with the SUBSEP character separating the subscripts.

The statement

```
break
```

exits a for or a while loop immediately.

```
continue
```

stops the current iteration of a for or while loop and begins the next iteration (if there is one).

```
next
```

terminates any processing for the current input record and immediately starts processing the next input record. Processing for the next record begins with the first appropriate rule.

```
exit[ (expr) ]
```

immediately goes to the END action if it exists; if there is no END action, or if **awk** is already performing the END action, the **awk** program terminates. **awk** sets the exit status of the program to the numeric value of *expr*. If you omit (*expr*), the exit status is 0.

```
return [expr]
```

returns from the execution of a function. If you specify an *expr*, the function returns the value of the expression as its result; otherwise, the function result is undefined.

```
delete array[i]
```

deletes element *i* from the given *array*.

```
print expr, expr, ...
```

is described in the *Output* subsection.

```
printf fmt, expr, expr, ...
```

is also described in the *Output* subsection.

Output

The `print` statement prints its arguments with only simple formatting. If it has no arguments, it prints the current input record in its entirety. **awk** adds the output record separator `ORS` to the end of the output that each `print` statement produces; when commas separate arguments in the `print` statement, the output field separator `OFS` separates the corresponding output values. `ORS` and `OFS` are built-in variables, the values of which you can change by assigning them strings. The default output record separator is a newline and the default output field separator is a space.

The variable `OFMT` gives the format of floating point numbers output by `print`. By default, the value is `%.6g`; you can change this by assigning `OFMT` a different string value. `OFMT` only applies to floating point numbers (ones with fractional parts).

The `printf` statement formats its arguments using the *fmt* argument. Formatting is the same as for the built-in function `sprintf`. Unlike `print`, `printf` does not add output separators automatically. This gives the program more precise control of the output.

The `print` and `printf` statements write to the standard output. You can redirect output to a file or pipe as described later.

If you add `>expr` to a `print` or `printf` statement, **awk** treats the string value of *expr* as a file name, and writes output to that file. Similarly, if you add `>>expr`, **awk** appends output to the current contents of the file. The distinction between `>` and `>>` is only important for the first `print` to the file *expr*. Subsequent outputs to an already open file append to what is there already.

To eliminate ambiguities, statements such as

```
print a > b c
```

are syntactically illegal. Use parentheses to resolve the ambiguity.

If you add `|expr` to a `print` or `printf` statement, **awk** treats the string value of *expr* as an executable command and runs it with the output from the statement piped as input into the command.

As mentioned earlier, you can have only a limited number of files and pipes open at any time. To avoid going over the limit, use the `close` function to close files and pipes when you no longer need them.

`print` and `printf` are also available as functions with the same calling sequence, but no redirection.

EXAMPLES

```
awk '{print NR ":" $0}' input1
```

outputs the contents of the file `input1` with line numbers prepended to each line.

The following is an example using `var=value` on the command line.

```
awk '{print NR SEP $0}' SEP=":" input1
```

awk can also read the program script from a file as in the command line:

```
awk -f addline.awk input1
```

which produces the same output when the file `addline.awk` contains

```
{print NR ":" $0}
```

The following program appends all input lines starting with `January` to the file `jan` (which may or may not exist already), and all lines starting with `February` or `March` to the file `febmar`:

```
/^January/ {print >> "jan"}
/^February|^March/ {print >> "febmar"}
```

This program prints the total and average for the last column of each input line:

```
      {s += $NF}
END   {print "sum is", s, "average is", s/NR}
```


The next program interchanges the first and second fields of input lines:

```
{
    tmp = $1
    $1 = $2
    $2 = tmp
    print
}
```

The following inserts line numbers so that output lines are left-aligned:

```
{printf "%-6d: %s\n", NR, $0}
```

The following prints input records in reverse order (assuming sufficient memory):

```
{
    a[NR] = $0 # index using record number
}
END {
    for (i = NR; i>0; --i)
        print a[i]
}
```

The next program determines the number of lines starting with the same first field:

```
{
    ++a[$1] # array indexed using the first field
}
END { # note output will be in undefined order
    for (i in a)
        print a[i], "lines start with", i
}
```

The following program can be used to determine the number of lines in each input file:

```
{
    ++a[FILENAME]
}
END {
    for (file in a)
        if (a[file] == 1)
            print file, "has 1 line"
        else
            print file, "has", a[file], "lines"
}
```

The following program illustrates how you can use a two dimensional array in **awk**. Assume the first field of each input record contains a product number, the second field contains a month number, and the third field contains a quantity (bought, sold, or whatever). The program generates a table of products versus month.

```
BEGIN {NUMPROD = 5}
{
    array[$1,$2] += $3
}
END {
    print "\t Jan\t Feb\tMarch\tApril\t May\t" \
        "June\tJuly\t Aug\tSept\t Oct\t Nov\t Dec"
    for (prod = 1; prod <= NUMPROD; prod++) {
        printf "%-7s", "prod#" prod
        for (month = 1; month <= 12; month++){
            printf "\t%5d", array[prod,month]
        }
        printf "\n"
    }
}
```

As the following program reads in each line of input, it reports whether the line matches a pre-determined value:

```
function randint() {
    return (int((rand()+1)*10))
}
BEGIN {
    prize[randint(),randint()] = "$100";
    prize[randint(),randint()] = "$10";
    prize[1,1] = "the booby prize"
}
{
    if (($1,$2) in prize)
        printf "You have won %s!\n", prize[$1,$2]
}
```

The following example prints lines, the first and last fields of which are the same, reversing the order of the fields:

```
$1==$NF {
    for (i = NF; i > 0; --i)
        printf "%s", $i (i>1 ? OFS : ORS)
}
```

The following program prints the input files from the command line. The `infiles` function first empties the passed array, and then fills the array. Notice that the extra parameter `i` of `infiles` is a local variable.

```
function infiles(f,i) {
    for (i in f)
        delete f[i]
    for (i = 1; i < ARGV; i++)
        if (index(ARGV[i], "=") == 0)
            f[i] = ARGV[i]
}
BEGIN {
    infiles(a)
    for (i in a)
        print a[i]
    exit
}
```

Here is the standard recursive factorial function:

```
function fact(num) {
    if (num <= 1)
        return 1
    else
        return num * fact(num - 1)
}
{ print $0 " factorial is " fact($0) }
```

The following program illustrates the use of `getline` with a pipe. Here, `getline` sets the current record from the output of the `wc` command. The program prints the number of words in each input file.

```
function words(file, string) {
    string = "wc " fn
    string | getline
    close(string)
    return ($2)
}
BEGIN {
    for (i=1; i<ARGC; i++) {
        fn = ARGV[i]
        printf "There are %d words in %s.",
            words(fn), fn
    }
}
```

ENVIRONMENT VARIABLES

`PATH` contains a list of directories that **awk** searches when looking for commands run by `system(expr)`, or input and output pipes.

Any other environment variable may be accessed by the **awk** program itself.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

When an **awk** program terminates because of a call to `exit()`, the exit status is the value passed to `exit()`.

Messages

- Message:** array "*name*" cannot be used as a scalar
Cause: You attempted to use the array *name* which has been used earlier in the script as a scalar. A variable can be used as an array or a scalar but not as both.
Action: Make sure that you use *name* as either a scalar or an array but not as both.
- Message:** attempt to redefine builtin function
Cause: You attempted to redefine one of the built-in **awk** functions.
Action: Choose a name for the function you are defining that is not the same as any of the built-in functions. See the *DESCRIPTION* section of this man page for lists of built-in arithmetic and string functions.
- Message:** cannot assign to function "*funcname*"
Cause: "*funcname*" is defined to be a function in your script and cannot be used as a variable.
Action: Use a different name for the variable.
- Message:** cannot open input file "*filename*"
Cause: **awk** was unable to open one of the files named on the command line.
Action: Check that the file exists, was named properly and that you have the appropriate permissions.
- Message:** cannot open script file "*filename*"
Cause: **awk** was unable to open one of the script files specified with the **-f** option.
Action: Check that the file exists, was named properly and that you have the appropriate permissions.
- Message:** division (/ or %) by zero
Cause: An arithmetic operation using / or % resulted in an attempt to divide by zero.
Action: Modify your program so that division by zero does not occur.
- Message:** EOF in regular expression
Cause: **awk** encountered the end-of-file character while reading a regular expression from the script file.
Action: Check for missing / delimiters at the end of regular expressions.
- Message:** EOF in string
Cause: **awk** encountered the end-of-file character while reading a string constant from the script file.
Action: Check for missing " delimiters at the end of string constants.
- Message:** error in function *funcname*(*arg*) at NR=*num*
Cause: A math error occurred while performing the function *funcname* on argument *arg*.
Action: Make sure that you are passing a proper argument to the function *funcname*.

-
- Message:** function "*funcname*" nesting level > *number*
Cause: There have been too many nested or recursive function calls. **awk** allows a maximum of *number* levels.
Action: Make sure that nested and recursive function calls do not exceed *number* levels of nesting.
- Message:** function "*funcname*" redefined
Cause: You attempted to redefine an existing function.
Action: Choose a new name for your function that does not conflict with any other function name.
- Message:** inadmissible use of reserved keyword
Cause: You attempted to use a reserved word in an unacceptable way such as a function or variable name.
Action: Choose a different name for your function or variable.
- Message:** insufficient arguments to printf or sprintf
Cause: You did not specify enough arguments to match the number required by the specified format string.
Action: Check your format string and number of arguments.
- Message:** insufficient memory for string storage
Cause: There were not enough free system resources for **awk** to use for string storage.
Action: Free up more system resources, or modify your **awk** program to require less string storage.
- Message:** invalid character "*char*" (hex *hexnum*)
Cause: **awk** encountered the invalid character *char* while processing the input file.
Action: Check the input file for invalid characters.
- Message:** lvalue required in assignment
Cause: You did not specify a variable or array element as the left-hand side of an assignment expression.
Action: Make sure that you specify a valid variable or array index on the left side of an assignment operator.
- Message:** may delete only array element or array
Cause: You attempted to use the **delete** statement to delete a scalar variable.
Action: Only use **delete** to delete arrays and array element.
- Message:** Missing field separator
Cause: You specified the **-F** option but did not follow it with a field separator.
Action: Provide a field separator following the **-F** option.

-
- Message:** Missing script file
Cause: You specified the **-f** option but did not follow it with the name of a script file.
Action: Provide the name of a script file following the **-f** option.
- Message:** Missing variable assignment
Cause: You specified the **-v** option but did not follow it with a variable assignment.
Action: Provide a variable assignment following the **-v** option.
- Message:** Newline in regular expression
Cause: **awk** encountered a newline while reading a regular expression.
Action: Check for a missing / delimiter.
- Message:** Newline in string
Cause: **awk** encountered a newline while reading a string constant.
Action: Check for a missing " delimiter.
- Message:** panic: sprintf() string longer than *number* characters
Cause: The maximum length of a string created by `sprintf()` is limited to *number* characters.
Action: Try processing the string in a different way.
- Message:** Record too long (LIMIT: *number* bytes)
Cause: **awk** read a record that was longer than the maximum record size it can handle. On UNIX and POSIX-compliant systems, the maximum record length is 20000 characters.
Action: Edit the offending record so that it does not exceed the limit.
- Message:** regular expression error
Cause: An error occurred while processing a regular expression.
Action: Check the regular expression.
- Message:** return outside of a function
Cause: **awk** encountered a **return** statement that is not part of a function.
Action: Only use the **return** statement inside a function definition.
- Message:** scalar "*name*" cannot be used as array
Cause: You attempted to use *name* as an array variable when it has already been used as a scalar.
Action: Make sure that you use a variable as either an array or a scalar, but not as both.
- Message:** second parameter to "split" must be an array
Cause: You invoked the `split` function but the second parameter was not an array.
Action: Ensure that `split` is invoked with an array as the second parameter.
- Message:** strcoll error, cannot malloc space.
Cause: There are not enough free system resources to allocate string space.
Action: Free up more resources.

-
- Message:** SYMTAB must have exactly one index
Cause: You tried to reference the SYMTAB array using more than one index.
Action: Always reference SYMTAB with exactly one index.
- Message:** syntax error "*regular expression error*" in */line/*
Cause: See **regerror**(3).
Action: See **regerror**(3).
- Message:** too deeply nested for in loop (*LIMIT: number*)
Cause: For loops can only be nested *number* levels deep.
Action: Re-write the script to use fewer levels.
- Message:** Too many fields (*LIMIT: number*)
Cause: **awk** read a record with more fields than it was able to handle.
Action: Edit the input file to decrease the number of fields in the offending record.
- Message:** too many open streams to *funcname* onto "*filename*"
Cause: **awk** can only have a limited number of files open at one time. There were too many open files.
Action: Make sure that unused files are being closed properly. If this doesn't fix the problem, restructure your program.
- Message:** unbalanced *char*
Cause: An unbalanced number of parentheses or braces was encountered.
Action: Make sure that all braces and parentheses are matched up.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **awk**.
Action: Check the *DESCRIPTION* of this man page for a list of valid **awk** options.
- Message:** unredirected *getline* in END action
Cause: The default input stream has already been closed by the time that the END action is performed so a *getline* which has not been redirected will fail.
Action: Redirect *getline* to read from a named file.
- Message:** variable "*name*" cannot be used as a function
Cause: You attempted to use the variable *name* as a function when it has not explicitly been defined as one, or when it has not been defined at all.
Action: Replace the offending variable *name* with the name of a function or define a function with that name.
- Message:** wrong number of arguments to function "*funcname*"
Cause: You attempted to invoke the function *funcname* with the wrong number of arguments.
Action: Specify the correct number of arguments for *funcname*.

LIMITS

Most constructions in this implementation of **awk** are dynamic, limited only by memory restrictions of the target machine. The parser stack depth is limited to 150 levels. Attempting to process extremely complicated programs may result in an overflow of this stack, causing an error.

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

The `ord` function is an extension to traditional implementations of **awk**. The `toupper` and `tolower` functions and the `ENVIRON` array are in POSIX and the UNIX System V Release 4 version of **awk**. This version is a superset of New AWK as described in *The AWK Programming Language* by Aho, Weinberger, and Kernighan.

The shell that the `system` function uses and that **awk** uses to run pipelines for `getline`, `print` and `printf` is system dependent. On the MPE/iX system, this is always the MPE/iX Shell.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

`ed(1)`, `egrep(1)`, `sed(1)`, `vi(1)`, `ascii(3)`, `regexp(3)`

NAME

banner — display text in large font

SYNOPSIS

banner [-c *char*] [-f *fontfile*] [-w *n*] [*text...*]

DESCRIPTION

banner writes the *text* arguments to the standard output in large letters using a default font. When no *text* arguments are present, **banner** reads text from the standard input.

Options

banner accepts the following options:

- c *char*
uses the single character *char* to form output characters instead of the default x.
- f *fontfile*
uses the output font from *fontfile* instead of the default font.
- w *n* limits the output width to at most *n* characters.

FILES

banner uses the following files:

- /etc/small.fnt
optional small font file.
- /etc/italic.fnt
optional italic font file.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** bad HALO font file format in "*fontfile*"
- Cause:** You specified an invalid font file.
- Action:** Use one of the files listed in the *FILES* section.

- Message:** cannot open font file "*fontfile*": *system error*
- Cause:** See **syserror(3)**.
- Action:** See **syserror(3)**.

-
- Message:** Missing font file
Cause: You specified the **-f** option but did not provide the name of a font file following it.
Action: Provide a font file following the **-f** option.
- Message:** Missing fill character
Cause: You specified the **-c** option but did not provide a character following it.
Action: Provide a character following the **-c** option.
- Message:** Missing width after **-w**
Cause: You specified the **-w** option but did not provide a width following it.
Action: Provide a width following the **-w** option.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **banner**.
Action: Check the *DESCRIPTION* section for a list of valid **banner** options.

PORTABILITY

x/OPEN Portability Guide 4.0. All UNIX systems.

The **-c**, **-f**, and **-w** options are extensions to traditional implementations of **banner**.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

NAME

basename — display file name component of path name

SYNOPSIS

basename *name* [*suffix*]

DESCRIPTION

basename strips off the leading part of a path name, leaving only the final component of the name, which is assumed to be the file name. To accomplish this, **basename** first checks to see if *name* consists of nothing but slash (/) characters. If so, **basename** replaces *name* with a single slash and the process is complete. If not, **basename** proceeds to remove any trailing slashes. If slashes still remain, **basename** strips off all leading characters up to and including the final slash. Finally, if you specify *suffix* and the remaining portion of *name* contains a suffix which matches *suffix*, then **basename** removes that suffix.

EXAMPLES

The command

```
basename src/dos/printf.c .c
```

produces

```
printf
```

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - unknown command line option
 - incorrect number of arguments

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

MPE/iX NOTES

On MPE/iX, **basename** is available as both a built-in shell utility and an external utility.

SEE ALSO

dirname(1)

NAME

bc — arbitrary-precision arithmetic calculation language

SYNOPSIS

bc [-i] [-l] [file ...]

DESCRIPTION

bc is a programming language which can perform arithmetic calculations to arbitrary precision. You can use it interactively, by entering instructions from the terminal. It can also run programs taken from files.

If you specify *file* arguments on the command line, they should be text files containing **bc** instructions. **bc** performs the instructions from those files, in the order that they appear on the command line, and then performs instructions from the standard input. **bc** terminates when it receives a `quit` instruction or reaches the end-of-file on standard input.

Options

bc accepts the following options.

- i puts **bc** into interactive mode. In this mode, **bc** displays a prompt when waiting for input. In addition, it handles errors somewhat differently. Normally, when **bc** encounters an error while processing a file, the interpreter displays the error message and exits. In interactive mode, the interpreter displays the message and returns to the prompt mode to allow debugging.
- l loads a library of standard mathematical functions before processing any other input. This library also sets the *scale* to 20. For a description of the functions in the `-l` library, see *Built-in Functions*.

The bc Language

bc is a simple but complete programming language with a syntax reminiscent of the C programming language. This version of **bc** is a superset of the standard language available on most systems. It has a number of additional features intended to make the language more flexible and useful. Features which are unique to this implementation are noted in the text.

Input consists of a series of instructions that assign values to variables or make calculations. It is also possible to define subprograms called *functions* which perform a sequence of instructions to calculate a single value.

bc displays the result of any line that calculates a value, but does not assign it to a variable. For example, the instruction

```
2+2
```

displays

```
4
```

By default, **bc** displays the result of any evaluated instruction followed by a newline. **bc** also saves the last value displayed in a special variable `.` so that you can use it in subsequent calculations.

Numbers

Numbers consist of an optional minus (-) sign followed by a sequence of zero or more digits, followed by an optional decimal point (.), followed by a sequence of zero or more digits. Valid digits are 0 through 9, and the hexadecimal digits A through F. The uppercase letters represent the values from 10 through 15. There must be at least one digit, either before or after the decimal point. If not, **bc** interprets the decimal point as the special variable `.` mentioned earlier.

A number can be arbitrarily long and may contain spaces. Here are some valid numbers with an input base of 10:

```
0    0.    .0    -3.14159    +09.    -12    1 000 000
```

Here are some valid numbers with an input base of 16 (`ibase=16`):

```
0    FF    FF.3    -10.444    A1
```

See *Bases* for more information.

A final point is that you cannot break up numbers with commas; you can write 1000000 or 1 000 000, but 1,000,000 results in an error message.

Identifiers

Identifiers are used as names for variables, functions, or arrays. Valid identifiers may include sequences containing any number of letters, digits or the underscore (_) character, but must start with a lowercase letter. Spaces are not allowed in identifiers. The ability to use identifiers more than one character in length is an extension not found in traditional implementations of **bc**.

- A variable holds a single numeric value. You can declare variables as local to a function using the `auto` statement (see *Functions*). All other variables are *global* and can be used anywhere. You do not need to declare global variables. **bc** creates variables as it requires them, with an initial value of zero. (Remember that there is also the special variable `.` (dot) which contains the result of the last calculation.)
- A function is a sequence of instructions that calculates a single value. A list of zero or more values enclosed in parentheses always follow a function name, as in `my_func(3.14159)`. See *Functions* later in this man page.

- An array is a list of values. Values in the list are called *elements* of the array. Each element in an array is numbered, beginning at zero. Such a number is known as a *subscript* or *index* of the array. Subscripts always appear in square brackets after the array. For example, `a[0]` refers to element zero in the array `a`. If a subscript value is a floating point number, the fractional part is discarded to make the subscript into an integer. For example, the following expressions all refer to the same element:

```
a[3]    a[3.2]    a[3.999]
```

The maximum number of elements in a **bc** array is given by the configuration variable `{BC_DIM_MAX}`. The valid array subscripts range from 0 to -1 inclusive. Unlike many languages, you don't need to declare the size of an array. Elements are created dynamically as required, with an initial value of zero.

Since parentheses always follow function names and square brackets always follow array names, **bc** can distinguish between the three types of names. Therefore, you can have variables, functions, and arrays with the same name. For example, `foo` may be a variable, while `foo()` is a function and `foo[]` is an array.

Built-In Variables

bc has a number of built-in variables which are used to control various aspects of the interpreter. These are described in the following sections.

Scale

The *scale* value is the number of digits to be retained after the decimal point in arithmetic operations. For example, if the scale is three, each calculation retains at least three digits after the decimal point. This means that

```
5 / 3
```

has the value

```
1.666
```

If `-1` is specified, the scale is set to 20; otherwise the default scale is zero.

The variable `scale` holds the current scale value. To change scales, assign a new value to `scale`, as in

```
scale = 5
```

Since `scale` is just a regular **bc** variable, it can be used in the full range of **bc** expressions.

The number of decimal places in the result of a calculation is affected not only by the scale, but also by the number of decimal places in the operands of the calculation. This is discussed in detail in the *Operations* section.

There is also a function `scale()` which can determine the scale of any expression. For example,

```
scale(1.1234)
```

returns the result four, which is the scale of the number 1.1234. The result of the `scale()` function is always an integer (that is, it has a scale of 0).

The maximum value for `scale` is given by the configuration variable `{BC_SCALE_MAX}` and the minimum value is 0.

Bases

bc lets you specify numbers in different bases, for example, octal (base 8) or hexadecimal (base 16). You can input numbers in one base and output them in a different base, simplifying the job of converting from one base to another. **bc** does this using the built-in variables `ibase` and `obase`

`ibase` is the base for input numbers. It has an initial value of 10 (normal decimal numbers). To use a different base for inputting numbers, assign an integer to `ibase`, as in

```
ibase = 8
```

This says that all future input numbers will be in base 8 (octal). The largest valid input base is 16 and the smallest valid input base is 2. Since there is no mechanism provided to represent digits larger than 15, bases larger than 16 are essentially useless. When the base is greater than 10, use the uppercase letters as digits. For example, base 16 uses the digits 0 through 9, and A through F. The digits are allowed in any number, regardless of the setting of `ibase` but are largely meaningless if the base is smaller than the digit. The one case where this is useful is in resetting the input base to 10. The constant A always has the value 10 no matter what `ibase` is set to, so to reset the input base to 10, type

```
ibase = A
```

`obase` is the base in which numbers are output. It has an initial value of 10 (normal decimal numbers). To change output bases, assign an appropriate integer to `obase`.

If the output base is 16 or less, **bc** displays numbers with normal digits and hexadecimal digits (if needed). The output base can also be greater than 16, in which case each *digit* is displayed as a decimal value and digits are separated by a single space. For example, if `obase` is 1000, the decimal number 123456789 is displayed as

```
123 456 789
```

Here, the *digits* are decimal values from 0 through 999. As a result, all output values are broken up into one or more *chunks* with three digits per chunk. Using output bases that are large powers of 10, you can columnate your output; for example, many users find that 100000 makes a good output base because numbers are grouped into chunks of five digits each.

Long numbers are output with a maximum of 70 characters per line. If a number is longer than this, **bc** puts a backslash (\) at the end of the line, indicating that the number is continued on the next line.

Internal calculations are performed in decimal, regardless of the input and output bases. Therefore, the number of places after the decimal point are dictated by the scale when numbers are expressed in decimal form.

The maximum value for `obase` is given by the configuration variable `{BC_BASE_MAX}`.

Arithmetic Operations

bc provides a large number of arithmetic operations. Following standard arithmetic conventions, some operations are calculated before others; for example, multiplication take place before addition unless you use parentheses to group operations. Operations that take place first are said to have a higher *precedence* than operations which take place later.

Operations also have an *associativity*. The associativity dictates the order of evaluation when you have a sequence of operations with equal precedence. Some operations are evaluated left to right while others are evaluated right to left. The *Operator Associativity* table shows the operators of **bc** from highest precedence to lowest. Programmers familiar with C will note that **bc**'s order of precedence is not the same as C's. In C, assignment operators have the lowest precedence. The precedence is shown in Table 1-3, *bc Operators*.

Operator	Associativity
()	left to right
unary ++ --	not applicable
unary - !	not applicable
^	right to left
* / %	left to right
+ -	left to right
= ^= *= /= %= +=	right to left
== <= >= != < >	none
&&	left to right
	left to right

Table 1-3: **bc** Operators

The following list describes each operator. In the descriptions, A and B can be numbers, variables, array elements, or other expressions. V must be either a variable or an array element.

- (A) An expression in parentheses is evaluated before any other operations are performed on it.

-A	is the negation of the expression.
!A	is the logical complement of the expression. If A evaluates to zero, !A evaluates to one. If A is not zero, !A evaluates to zero. This operator is unique to this version of bc .
++V	adds 1 to the value of V. The result of the expression is the new value of V.
--V	subtracts 1 from the value of V. The result of the expression is the new value of V.
V++	adds 1 to the value of V, but the result of the expression is the old value of V.
V--	subtracts 1 from the value of V, but the result of the expression is the old value of V.
A ^ B	calculates A to the power B. B must be an integer. The scale of the result of A^B is <pre>min(scale(A) * abs(B), max(scale, scale(A)))</pre> where <code>min()</code> calculates the minimum of a set of numbers and <code>max()</code> calculates the maximum.
A * B	calculates A multiplied by B. The scale of the result is <pre>min(scale(A) + scale(B), max(scale, scale(A), scale(B)))</pre>
A / B	calculates A divided by B. The scale of the result is the value of <code>scale</code> .
A % B	calculates the remainder from the division of A by B. This is calculated in two steps. First, bc calculates A/B to the current scale. It then obtains the remainder through the formula $A - (A / B) * B$ calculated to the scale <pre>max(scale + scale(B), scale(A))</pre>
A + B	adds A plus B. The scale of the result is the maximum of the two scales of the operands.
A - B	calculates A minus B. The scale of the result is the maximum of the two scales of the operands.

The next group of operators are all *assignment* operators. They assign values to objects. An assignment operation has a value: the value that is being assigned. Therefore you can write operations like `a=1+(b=2)`. In this operation, the value of the assignment in parentheses is 2 because that is the value assigned to `b`. Therefore, the value 3 is assigned to `a`. The possible assignment operators are:

`V = B` assigns the value of `B` to `V`.
`V ^= B` is equivalent to `V=V^B`.
`V *= B` is equivalent to `V=V*B`.
`V /= B` is equivalent to `V=V/B`.
`V %= B` is equivalent to `V=V%B`.
`V += B` is equivalent to `V=V+B`.
`V -= B` is equivalent to `V=V-B`.

The following expressions are called *relations*, and their values can be either true (one) or false (zero). This version of **bc** lets you use the relational operators in any expression, not just in the conditional parts of `if`, `while`, or `for` statements. These operators work in exactly the same way as their equivalents in the C language. The result of a relation is zero if the relation is false and one if the relation is true.

`A == B` is true if and only if `A` equals `B`.
`A <= B` is true if and only if `A` is less than or equal to `B`.
`A >= B` is true if and only if `A` is greater than or equal to `B`.
`A != B` is true if and only if `A` is not equal to `B`.
`A < B` is true if and only if `A` is less than `B`.
`A > B` is true if and only if `A` is greater than `B`.
`A && B` is true if and only if `A` is true (non-zero) and `B` is true. If `A` is not true, the expression `B` is never evaluated.
`A || B` is true if `A` is true or `B` is true. If `A` is true, the expression `B` is never evaluated.

Comments and White Space

A *comment* has the form

```
/* Any string */
```

Comments can extend over more than one line of text. When **bc** sees `/*` at the start of a comment, it discards everything up to the next `*/`. The only effect a comment has is to indicate the end of a token.

As an extension, this version of **bc** also provides an additional comment convention using the `#` character. All text from the `#` to the end of the current line is treated as a single blank, as in

```
2+2 # this is a comment
```

bc is free format. You may freely insert blanks or horizontal tab characters to improve the readability of the code. Instructions are assumed to end at the end of the line. If you have an instruction that is so long you need to continue it onto a new line, put a backslash (`\`) as the last character of the first line and continue the instruction on the next line. For example,

```
a = 2\  
+ 3
```

This is equivalent to

```
a = 2 + 3
```

Instructions

A **bc** instruction may be an expression that performs a calculation, an assignment, a function definition, or a statement. If an instruction is not an assignment, **bc** displays the result of the instruction when it has completed the calculation. For example, if you enter

```
3.14 * 23
```

bc displays the result of the calculation. However, with

```
a = 3.14 * 23
```

bc does not display anything because the expression is an assignment. If you do want to display the value of an assignment expression, place the expression in parentheses.

The following list shows the instruction formats recognized by **bc**.

expression
calculates the value of the *expression*.

"string"

is a string constant. When **bc** sees a statement with this format, it displays the contents of the string. For example,

```
"Hello world!"
```

tells **bc** to display `Hello world!` A newline character is *not* output after the string. This makes it possible to do things like

```
foo = 15
"The value of foo is "; foo
```

With these instructions, **bc** displays

```
The value of foo is 15
```

statement ; statement ...

is a sequence of statements on the same line. In **bc**, a semicolon (`;`) is equivalent to a newline. They both indicate the end of a statement. **bc** performs these statements from left to right.

{ statement }

is a brace-bracketed statement. Brace brackets are used to group sequences of statements together, as in

```
{
  statement
  statement
  ...
}
```

Brace brackets can group a series of statements which are split over several lines. They are usually used with control statements like `if` and `while`.

`break`

can only be used inside a `while` or `for` loop. `break` terminates the loop.

`for (initexp ; relation ; endexp) statement`

is equivalent to

```
initexp
while (relation) {
  statement
  endexp
}
```

where *initexp* and *endexp* are expressions and *relation* is a relation. For example,

```
a = 0
for (i = 1; i <= 10; ++i) a += i
```

is equivalent to the `while` example given earlier. C programmers should note that all three items inside the parentheses must be specified; unlike C, `bc` does not let you omit any of these expressions.

`if (relation) statement`

tests whether the given *relation* is true. If it is, `bc` performs the *statement*; otherwise, `bc` skips over *statement* and goes to the next instruction. For example,

```
if ((a%2) == 0) "a is even"
```

displays `a is even` if `a` has an even value.

`if (relation) statement1 else statement2`

is similar to the simple `if` statement. If *relation* is true, it performs *statement1*; otherwise, it performs *statement2*. It may be used as follows:

```
if ((a%2) == 0) "a is even" else "a is odd"
```

Note: There is no statement separator between "a is even" and the `else` keyword. This differs from the C language.

Here is another example:

```
if (a<10) {
    "a "
    "is "; "less than 10 "
    a
} else {
    "a is"
    " greater than 10 "
    a
}
```

Note: The braces must be on the same line as the `if` and the `else` keywords. This is because a newline or a semicolon right after *(relation)* indicates that the body of the statement is null. One common source of errors in `bc` programs is typing the statement portion of an `if` statement on a separate line. If `-i` is used, the interpreter displays a warning when `if` statements with null bodies are encountered.

`while (relation) statement`
repeatedly performs the given *statement* while *relation* is true. For example,

```
i = 1
a = 0
while (i <= 10) {
    a += i
    ++i
}
```

adds the integers from 1 through 10 and stores the result in `a`.

If the *relation* is not true when `bc` encounters the while loop, `bc` does not perform *statement*.

`print expression , expression ...`

displays the results of the *expressions*. Normally `bc` displays the value of each expression or string it encounters. This makes it difficult to format your output in programs. For this reason, the MPE/iX Shell and Utilities version of `bc` has a `print` statement to give you more control over how things are displayed. `print` lets you display several numbers on the same line with strings. This statement displays all of its arguments on a single line. A single space is displayed between adjacent numbers (but not between numbers and strings). A `print` statement with no arguments displays a newline. If the last argument is null, subsequent output continues on the same line. Here are some examples of how to use `print`:

```
/* basic print statement */
print "The square of ", 2, "is ", 2*2
The square of 2 is 4

/* inserts a space between adjacent numbers */
print 1,2,3
1 2 3

/* note - no spaces */
print 1,"",2,"",3
123

/* just print a blank line */
print

/* two statements with output on same line */
print 1,2,3, ; print 4, 5, 6
1 2 3 4 5 6
```

`quit`

terminates **bc**. In other implementations of **bc**, the interpreter exits as soon as it reads this token. This version of **bc** treats `quit` as a real statement, so you can use it in loops, functions, and so on.

`sh ...`

lets you send a line to the MPE/iX Shell for execution, as in

```
sh more <foo
```

This command passes everything from the first non-blank character until the end of the line to the shell for execution.

`void expression`

`void` throws away or *voids* the result of the evaluation of *expression* instead of displaying it. This is useful when using `++` and `--` operators, or when you want to use a function but don't want to use the return value for anything. For example,

```
void foo++
```

increments `foo` but does not display the result. The `void` statement is unique to this version of **bc**.

Several other types of statements are only relevant in function definitions. These are described in the next section.

Functions

A function is a *subprogram* to calculate a result based on *argument* values. For example, the following function converts a temperature given in Fahrenheit into the equivalent temperature in Celsius.

```
define f_to_c(f) {  
    return ((f-32) * 5 / 9)  
}
```

This defines a function named `f_to_c()` that takes a single argument called `f`. The *body* of the function is enclosed in brace brackets. The opening brace must be on the same line as the `define` keyword. The function body consists of a sequence of statements to calculate the *result* of the function. An expression of the form

```
return (expression)
```

returns the value of *expression* as the result of the function. The parentheses around the expression are optional.

To activate the subprogram you use a *function call*. This has the form

```
name ( expression [ , expression ] ... )
```

where *name* is the name of the function, and the *expressions* are argument values for the function. A function call can be used anywhere you might use any other expression. The value of the function call is the value that the function returns. For example, with the function `f_to_c()` described earlier, `f_to_c(41)` has the value 5 (since 41 Fahrenheit is equivalent to 5 Celsius).

The general form of a function definition is

```
define name([parameter][,parameter]...) {
    auto local, local, ...
    statement
    statement
    ...
}
```

The *parameters* on the first line may be variable names or array names. Array names are indicated by putting square brackets after them. For example, if `cmpvec()` is a function that compares two vectors, the function definition might start with

```
define cmpvec(a[],b[]) {
```

Parameters do not conflict with arrays or variables of the same name. For example, you may have a parameter named `a` inside a function, and a variable named `a` outside, and the two are considered entirely separate entities. Assigning a value to the variable does not change the parameter and vice versa. All parameters are *passed by value*. This means that a copy is made of the argument value and is assigned to the formal parameter. This also applies to arrays. If you pass an array to a function, a copy is made of the whole array, so any changes made to the array parameter don't affect the original array.

A function may not need any arguments. In this case, the `define` line does not have any parameters inside the parentheses, as in

```
define f() {
```

The `auto` statement declares a sequence of *local* variables. When a variable or array name appears in an `auto` statement, the current values of those items are saved and the items are initialized to zero. For the duration of the function, the items have their new values. When the function terminates, the old values of the items are restored. Note, however, that **bc** uses dynamic scoping rules, unlike C which uses lexical scoping rules (see the *NOTES* section for more details).

For example,

```
define addarr(a[],l) {
    auto i, s
    for (i=0; i < l; ++i) s += a[i]
    return (s)
}
```

is a function that adds the elements in an array. The argument `l` stands for the number of elements in the array. The function uses two local names: a variable named `i` and a variable named `s`. These variables are *local* to the function `addarr()` and are unrelated to objects of the same name outside the function (or in other functions). Objects named in an `auto` statement are called *autos*. Autos are initialized to zero each time the function is called. Thus the sum `s` is set to zero each time this function is called. You may also have local arrays, which are specified by placing square brackets after the array name in the `auto` statement.

```
define func_with_local_array() {
    auto local_array[];
    for(i=0; i<100; i++) local_array[i] = i*2
}
```

This example defines a local array called `local_array`. Local arrays start out with no elements in them.

If a function refers to an object that is not a parameter and not declared `auto`, the object is assumed to be *external*. External objects may be referred to by other functions or by statements which are outside of functions. For example,

```
define sum_c(a[],b[],l) {
    auto i
    for (i=0; i < l; ++i) c[i] = a[i] + b[i]
}
```

references an external array named `c` which is the element-by-element sum of two other arrays. If `c` did not exist prior to calling `sum_c()`, it is created dynamically. Once the program has called `sum_c()`, statements in the program or in functions can refer to array `c`.

Functions usually require a `return` statement. This has the form

```
return (expression)
```

The *expression* is evaluated and used as the result of the function. The expression must have a single numeric value; it cannot be an array.

A `return` statement terminates a function, even if there are more statements left in the function. For example,

```
define abs(i) {
    if (i < 0) return (-i)
    return (i)
}
```

is a function that returns the absolute value of its argument. If `i` is less than zero, the function takes the first `return`; otherwise, it takes the second.

A function can also terminate by performing the last statement in the function. If so, the result of the function is zero. The function `sum_c()` is an example of a function that doesn't have a `return` statement. The function doesn't need a `return` statement, because its work is to calculate the external array `c`, not to calculate a single value. Finally, if you want to return from a function, but not return a value you may use

```
return ( )
```

or simply

```
return
```

If there are no parameters to the `return` statement, a default value of zero is returned.

Built-In Functions

bc has a number of built-in functions that perform various operations. These functions are similar to user-defined functions with the exception that you don't have to define them yourself — they are already set up for you. These functions are:

`length(expression)`

calculates the total number of decimal digits in *expression*. This includes digits both before and after the decimal point. The result of `length()` is an integer. For example, `length(123.456)` returns 6.

`scale(expression)`

returns the scale of *expression*. For example, `scale(123.456)` returns 3. The result of `scale()` is always an integer. Subtracting the scale of a number from the length of a number lets you determine the number of digits before the decimal point.

`sqrt(expression)`

calculates the square root of the value of *expression*. The result is truncated in the least significant decimal place (not rounded). The scale of the result is the scale of *expression*, or the value of `scale()`, whichever is larger.

You can use the following functions if `-l` is specified on the command line. If it is not, the function names are not recognized. There are two names for each function: a full name, and a single character name for compatibility with POSIX.2. The full names are the same as the equivalent functions in the standard C math library.

`atan(expression)` or `a(expression)`

calculates the arctangent of *expression*, returning an angle in radians.

`cos(expression)` or `c(expression)`

calculates the cosine of *expression*, where *expression* is an angle in radians.

`exp(expression)` or `e(expression)`

calculates the exponential of *expression* (that is, the value *e* to the power of *expression*).

`jn(integer,expression)` or `j(integer,expression)`

calculates the Bessel function of *expression*, with order *integer*.

`log(expression)` or `l(expression)`

calculates the natural logarithm of *expression*.

`sin(expression)` or `s(expression)`

calculates the sine of *expression*, where *expression* is an angle in radians.

EXAMPLES

This sections provides some examples of how to use the `bc` language to accomplish various things.

Here is a simple function to calculate the sales tax on a purchase. The amount of the purchase is given by `purchase`, and the amount of the sales tax (in per cent) is given by `tax`.

```
define sales_tax(purchase,tax) {
    auto old_scale
    scale = 2
    tax = purchase*(tax/100)
    scale = old_scale
    return (tax)
}
```

For example,

```
sales_tax(23.99,6)
```

calculates 6% tax on a purchase of \$23.99. The function temporarily sets the scale value to 2 so that the monetary figures have two figures after the decimal point. Remember that `bc` truncates calculations instead of rounding, so some accuracy may be lost. It is better to use one more digit than needed and perform the rounding at the end. The `round2()` function, shown later in this section, rounds a number to two decimal places.

Division resets the scale of a number to the value of `scale`. This can be used as follows to extract the integer portion of a number.

```
define integer_part(x) {
    # a local to save the value of scale
    auto old_scale
    # save the old scale, and set scale to 0
    old_scale = scale; scale=0
    # divide by 1 to truncate the number
    x /= 1
    # restore the old scale
    scale=old_scale
    return (x)
}
```

Having defined this function, it is now trivial to define one to return the fractional part of a number.

```
define fractional_part(x) {
    return (x - integer_part(x))
}
```

The following function lets you set the scale of a number to a given number of decimal places.

```
define set_scale(x, s) {
    auto os
    os = scale
    scale = s
    x /= 1
    scale = os
    return (x)
}
```

`set_scale()` can now be used in a function which rounds a number to two decimal places.

```
define round2(num) {
    auto temp;
    if(scale(num) < 2) return (set_scale(num, 2))
    temp = (num - set_scale(num, 2)) * 1000
    if(temp > 5) num += 0.01
    return (set_scale(num,2))
}
```

This is a very useful function if you want to work with monetary values. For example, you can now rewrite `sales_tax()` to use `round2()`.

```
define sales_tax(purchase,tax) {
    auto old_scale
    scale = 2
    tax = round2(purchase*(tax/100))
    scale = old_scale
    return (tax)
}
```

Here is a function which recursively calculates the factorial of its argument.

```
define fact (x) {
    if(x < 1) return 1
    return (x*fact(x-1))
}
```

The factorial function can also be written iteratively as:

```
define fact (x) {
    auto result
    result = 1
    while(x>1) result *= x--
    return (result)
}
```

With either version, `fact(6)` returns 720.

Here is another recursive function. This one calculates the *n*th element of the Fibonacci sequence.

```
define fib(n) {
    if(n < 3) {
        return (1)
    } else {
        return (fib(n-1)+fib(n-2))
    }
}
```

FILES

bc uses the following file:

`/usr/lib/lib.b`

File containing the library of functions loaded with `-l`.

DIAGNOSTICS

Possible exit status values are:

0 Successful completion.

1 An error occurred.

Messages

Message: break statement found outside of loop

Cause: **bc** encountered a break statement when it was not performing a for or while loop.

Action: Make sure that all break statements occur within for or while loops.

Message: built-in *var* can't be used as a parameter or auto variable

Cause: You attempted to use the built-in variable *var* as a parameter or auto variable.

Action: Do not use built-in variables as parameters or auto variables.

Message: can't pass array to '*var*'

Cause: You attempted to pass an array to the scalar variable *var*.

Action: Make sure that the value which you pass to a scalar variable is a scalar.

Message: can't pass scalar to '*var[]*'

Cause: You attempted to pass a scalar to the array *var[]*.

Action: Make sure that the value which you pass to an array variable is an array.

Message: divide by 0

Cause: You attempted to divide by 0.

Action: Do not divide by 0.

Message: end of file in comment starting on line *num* of *filename*

Cause: **bc** encountered the end-of-file character when reading a comment which begins on line *num* of the file *filename*.

Action: Make sure that the file *filename* contains a `/*` to end each comment begun with a `*/`.

-
- Message:** end of file in string starting on line *num* of *filename*
Cause: **bc** encountered the end-of-file character when reading a string which begins on line *num* of the file *filename*.
Action: Make sure that the file *filename* contains a " to end each string.
- Message:** exponent must be an integer from 0 to *number*.
Cause: You specified an exponent that was not an integer in the range 0 to SHRT_MAX-1.
Action: Specify an exponent in the valid range.
- Message:** *filename* : system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** *funcname*() is not a function
Cause: You attempted to use a name that is not defined as a function in a function context.
Action: Specify a valid function name.
- Message:** numerical constant is too long
Cause: You specified a numerical constant that was longer than the maximum permitted length, as defined by the value of the configuration variable BC_STRING_MAX.
Action: Specify a shorter numerical constant.
- Message:** L?
Cause: You tried to pop a value off of an empty stack variable using the **L** operator.
Action: Correct your program.
- Message:** out of memory
Cause: **bc** ran out of system resources while trying to allocate space. If **bc** is being run interactively, it tries to free up more resources and returns to the top level of the interpreter.
Action: Free up some resources and try again. Pay particular attention to large arrays.
- Message:** out of memory (fatal)
Cause: **bc** ran out of system resources but was unable to recover sufficient storage to continue.
Action: Free up some resources and try again. Pay particular attention to large arrays.
- Message:** Q?
Cause: You specified a string argument to the **Q** command. This is invalid. The **Q** command requires a numeric argument.
Action: Correct your program.

-
- Message:** shell command failed to execute
Cause: You specified the `sh` statement with *command* as its argument and `bc` failed to run *command*.
Action: Check the syntax of the specified command.
- Message:** sqrt of negative number
Cause: You attempted to take the square root of a negative number.
Action: Only use the `sqrt` function with positive numbers.
- Message:** string is too long
Cause: You specified a string that was longer than the maximum permitted length, as defined by the value of the configuration variable `BC_STRING_MAX`.
Action: Specify a shorter string.
- Message:** syntax error
Cause: A syntax error was found.
Action: Correct the syntax error.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for `bc`.
Action: Check the *DESCRIPTION* section for a list of valid `bc` options.
- Message:** valid array index is 0 through *num*
Cause: You specified an array index that was not in the range 0 to `BC_DIM_MAX-1`, where `BC_DIM_MAX` is a configuration variable indicating the maximum number of elements that a `bc` array may have.
Action: Specify an array index in the indicated range.
- Message:** '*var*' can only have values from *num1* through *num2*
Cause: You attempted to assign a value to the variable *var* that was not in the range *num1* to *num2*.
Action: Only assign values in the range *num1* to *num2* to the variable *var*.
- Message:** warning: body of if/else statement is empty
Cause: You did not supply any statements for the body of an `if` or `if/else` construct. `bc` only generates this message when you have specified the `-i` option.
Action: Make sure that this is what you intended. Check the *DESCRIPTION* section for a discussion of `bc` syntax.
- Message:** warning: '==' operator assumed
Cause: This version of `bc` permits the use of the old style assignment operators like `==` rather than `=`. This can be ambiguous since `a=-2` can mean `a = - 2` or `a = -2`.
Action: Use spaces to clarify the syntax of the expression.

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

The `-i` option, the `&&` and `||` operators, the `if ... else ...` statement, the `print` statement, the `sh` statement, and the optional parentheses in the `return` statement are extensions to the POSIX standard.

NOTES

This section describes some additional details about `bc` that may be useful to know.

Unlike the C language which uses lexical scoping rules, `bc` uses dynamic scoping. This is most easily explained with an example:

```
a=10
define f1() {
    auto a;
    a = 13;
    return (f2())
}
define f2() {
    return (a)
}
f1()
13
f2()
10
```

If `f1()` is called, `bc` displays the number 13, instead of the number 10. This is because `f1()` hides away the old (global) value of `a` and then sets it to 13. When `f2()` refers to `a` it sees the variable dynamically created by `f1()` and so displays 13. When `f1()` returns, it restores the old value of `a`. When `f2()` is called directly, instead of through `f1()` it sees the global value for `a` and displays 10. The corresponding C code displays 10 in both cases.

Numbers are stored as strings in the program and converted into numbers each time they are used. This is important because the value of a *constant* number may change depending on the setting of the `ibase` variable. For example, suppose the following instructions are given to `bc`:

```
define ten() {
    return (10)
}
ten()
10
```

```
ibase=16
ten()
16
```

In this example, when the base is set to 10, `ten()` returns the decimal value 10; however, when the input base is changed to 16, the function returns the decimal value 16. This can be a source of confusing errors in `bc` programs.

Finally, the library of functions loaded using the `-l` option is stored in the file

```
/usr/lib/lib.b
```

under your root directory. This is a simple text file which you can examine and change to add new functions as desired.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

`dc(1)`

NAME

break — exit from loop in shell script

SYNOPSIS

break [*number*]

DESCRIPTION

break exits from a `for`, `select`, `until`, or `while` loop in a shell script. If *number* is given, **break** exits from the given number of enclosing loops. The default value of *number* is 1.

DIAGNOSTICS

break always exits with an exit status of zero.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh**(1) man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

break is a built-in command of the Bourne Shell and KornShell on UNIX systems.

NOTE

MPE/iX Shell and Utilities implements **break** as a special built-in command of the shell.

SEE ALSO

continue(1), **sh**(1)

NAME

c — produce multi-column output

SYNOPSIS

c [**-hvv**] [**-g** *gutterwidth*] [**-w** *pagewidth*] [*file* ...]

DESCRIPTION

The **c** command reads lines from each input *file*, or from standard input if no *file* is specified. **c** produces multi-column output from this input. It places as many input lines across the page as will fit in the prescribed width, including a gutter between adjacent columns.

Options

c accepts the following options:

-g *gutterwidth*

specifies a *gutterwidth* measured in characters. The default is two characters.

-h orders input lines from left to right, horizontally. This is the default.

-v balances the columns so that they are all the same length, or as close as possible.

-V orders input lines vertically down the columns.

-w *pagewidth*

specifies the width of the entire page, in characters. The default *pagewidth* is 80 characters. If you do not set the width with the **-w** option, **c** checks to see if there is an environment variable named *COLUMNS*. If so, **c** uses its value as the *pagewidth*, in characters. For example, if you are using the MPE/iX Shell,

```
export COLUMNS=132
```

sets a *pagewidth* of 132 characters.

EXAMPLES

Try the following to see the effect of these variants of the command:

```
ls /bin | c
ls /bin | c -v
ls /bin | c -V
```

ENVIRONMENT VARIABLES

c uses the following environment variable:

COLUMNS

used as the page width if you do not specify the **-w** option.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** file "*filename*": system error
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Missing gutter size
Cause: You specified the **-g** option without providing a value for *gutterwidth*.
Action: Provide a value for *gutterwidth* following the **-g** option.
- Message:** Missing width after **-w**
Cause: You specified the **-w** option without providing a value for *pagewidth*.
Action: Provide a value for *pagewidth* following the **-w** option.
- Message:** too much input for available memory
Cause: The files provided on the command are too large for **c** to handle with the system resources that are currently available.
Action: Free up more system resources or split the input files into smaller pieces and process them separately.
- Message:** Unknown option "*-option*"
Cause: You specified *option* as an option on the command line. This is not a valid option for the **c** command.
Action: See the *DESCRIPTION* section of this man page for a list of valid options.

PORTABILITY

Some UNIX systems.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

ls(1), **pr(1)**

NAME

c89 – generic C compiler interface

SYNOPSIS

```
c89 [-cEgOs] [-D name[=value]] ... [-I directory] ... [-L directory] ...
[-o output] [-P listfile] [-T stacksize] [-U name] ... [-Wphase,arg...] operand...
```

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

c89 is an interface to the C compiler and linker on your system. It accepts source code conforming to the C standard. The files specified by *operands* are compiled and linked to produce an executable file. By default, the executable file is written to `a.out`.

Options

c89 accepts the following options:

- c** suppresses the link-edit phase of the compilation and does not delete any object files that are produced. Each *operand* of the form *file.c* produces an object file named *file.o*.
- D name[=value]** defines *name* as if by a `#define` directive. If you omit *=value*, **c89** assumes a value of 1.
- E** copies the C source to the standard output after expanding all preprocessor directives. The source code is not compiled.
- g** produces symbolic information for use by debuggers in the object or executable files.
- I directory** adds *directory* to the search path for `#include` files which do not have absolute path names. If the name of an `#include` file is enclosed in double-quotes (" "), **c89** first searches the directory of the file containing the `#include` directive. It then searches the directories indicated by **-I** options and finally, it searches `/usr/include`. If the name of an `#include` file is enclosed in angle brackets (< >), **c89** first searches the directories indicated by **-I** options, and then `/usr/include`.
- L directory** adds *directory* to the search path for libraries. **c89** first searches the directories indicated by **-L** options in the order they appear on the command line, and then searches `/lib` and `/usr/lib`.

-
- O** optimizes the executable output.
 - o *output***
writes the executable output to the file *output*. If you specify the **-c** or **-E** option, no executable output is created.
 - P *listfile***
specifies a listing file for the C compile. It passes *listfile* to the file equation for `cclist`. At present, it is not possible to create a listing file using the POSIX name syntax. You must use traditional MPE/iX names with this option. For example,

```
c89 -c -P listing.mygroup.sys foo.c
```
 - T *stacksize***
passes *stacksize* to the link editor which uses it to set the native mode stack size of the program. This is equivalent to

```
-WL,nmstack=stacksize
```
 - s** produces *stripped* object and/or executable files from which symbolic and other information not required for proper execution using **exec** has been removed.
 - U *name***
undefines the symbol *name*. If *name* appears in both a **-D** and a **-U** option, it is undefined regardless of the order in which the options were specified.
 - W*phase,arg ...***
passes arguments to other phases of the compile process. *phase* can normally be one of two values:
 - c** passes the arguments (*arg ...*) to the C compiler.
 - l** passes the arguments (*arg ...*) to the linker.
 On MPE/iX, however, there are two ways to pass arguments to the link editor. As a result, *phase* can be one of the following three values:
 - c** passes the arguments to the compiler `CCOMXL.PUB.SYS`
 - l** places the arguments in the indirect file passed to the link editor `LINKEDIT.PUB.SYS`
 - L** adds the arguments to the command line of the link editor `LINKEDIT.PUB.SYS`

Here is an example of how these options might be used to compile a program that uses AIFs.

```
c89 "-Wl, -o AIF1" -Wc,+e "-WL,cap=ph,pm,ia;PRIVLEV=2" aif1.c
```


- Wl is used to set the output file name.
- WL is used to set the capabilities and privilege level of the program. It should be noted that if -WL is not specified, cap=ph is passed to the link editor by default. If you do specify -WL, this default is not passed.
- Wc is used to turn on the extension in the compiler that supports long pointers.

Operands

c89 accepts the following operands:

-l *library*

searches the library named *liblibrary.a*. c89 searches a library when its name is encountered, so the placement of -l operands is important.

-l c This library contains all functions defined by POSIX.1 as well as the ANSI C language bindings. This library is searched even if you do not specify it on the command line.

-l m refers to the library containing all functions referenced in <math.h>.

-l l refers to the library containing all functions required by the C output of **lex** that are not available through the -l c operand.

-l y refers to the library containing all functions required by the C output of **yacc** that are not available through the -l c operand.

file.a is a library of object files usually produced by **ar**(1), which is passed directly to the link editor.

file.c is a C source file to be compiled and optionally linked.

file.o is an object file created by c89 -c, which is passed directly to the link editor.

ENVIRONMENT VARIABLES

c89 uses the following environment variable:

TMPDIR

contains a path name that overrides the default directory for temporary files, if any.

FILES

c89 uses the following file:

a.out contains the executable output unless the -o *output* option is specified.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** cannot mix `-g` and `-s` options
Cause: You specified both the `-g` and `-s` options on the command line. These two options are mutually exclusive.
Action: Consult the *DESCRIPTION* section and correct the command line.
- Message:** cannot mix `-g` and `-O`
Cause: You specified both the `-g` and `-O` options on the command line. These two options are mutually exclusive.
Action: Consult the *DESCRIPTION* section and correct the command line.
- Message:** cannot mix `-s` and `-E`
Cause: You specified both the `-s` and `-E` options on the command line. These two options are mutually exclusive.
Action: Consult the *DESCRIPTION* section and correct the command line.
- Message:** compile failed
Cause: **c89** was unable to run the C compiler.
Action: Make sure that your *PATH* environment variable includes the directory containing the compiler, and that you have the necessary permissions to run the compiler.
- Message:** link failed
Cause: **c89** was unable to run the **linkedit** utility.
Action: Make sure that your *PATH* environment variable includes the directory containing the **linkedit** utility, and that you have the necessary permissions to run that utility.
- Message:** no files specified
Cause: You failed to specify a valid operand file name on the command line.
Action: Correct the command, making sure to specify operands.
- Message:** unknown file extension
Cause: You specified an operand file name with an unrecognized file extension. The recognized extensions are `.c`, `.a`, and `.o`.
Action: Correct the file name.

Message: unknown option *-option*
Cause: You specified an option that is not valid for **c89**.
Action: Check the *DESCRIPTION* section for a list of valid **c89** options.

For specific C compiler and link editor error messages, refer to Appendix A of the *HP C/iX Reference Manual* (Part Number 31506-90005) and to Appendix A of the *HP Link Editor/iX Reference Manual* (Part Number 32650-90030).

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

MPE/iX NOTES

The **c89** command is currently implemented on MPE/iX as a script-driven front end to the MPE/iX C compiler and link editor.

If the environment variable *ECHO* is defined, the **c89** utility displays the commands that are passed to the MPE/iX CI for execution. For example

```
$ ECHO=1 c89 -o myprog test.c
```

displays each of the commands submitted to the MPE/iX CI.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

ar(1), **strip(1)**

NAME

calendar — appointment reminder system

SYNOPSIS

calendar [-]

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

With no options, **calendar** displays all *current* appointments on the standard output. **calendar** searches the file `calendar` in the current directory, looking for lines that match either today's or tomorrow's date. On Friday, Saturday, or Sunday, *tomorrow* extends through to Monday. Each *appointment* must fit on a single line, with the date formatted as one of:

```
January 27
1/27
jan 27
```

Note: The month name may be abbreviated to three letters; the case is not significant; the month can be given numerically.

calendar accepts the following option:

- searches for a calendar file in each home directory found in the user database. **calendar** sends mail to the corresponding user for any appointments that are found to be current. Mail is sent using the **mailx** utility or alternatively the utility named in the *MAILER* environment variable.

EXAMPLE

If today is Friday April 7th, and the following `calendar` file is found in the current directory:

```
tue mar 7    1:00 pm    dentist
Sat April 8  Trip to the zoo
mon april 10    3:30 pm job interview
4/11         vacation starts
```

calendar prints the following:

```
Sat April 8  Trip to the zoo
mon april 10    3:30 pm job interview
```

ENVIRONMENT VARIABLES

MAILER

contains the name of the utility that **calendar** uses to send mail. If this variable is not set, **calendar** uses **mailx** as the default mail utility.

FILES

calendar

file used in the current directory, or user's home directory.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

Message: calendar file: *system error*

Cause: See **syserror**(3).

Action: See **syserror**(3).

Message: insufficient memory available

Cause: There were not enough free system resources for the operation being performed.

Action: Free up more system resources.

Message: internal RE error: *regular expression error*

Cause: See **regerror**(3).

Action: See **regerror**(3).

PORTABILITY

All UNIX systems. *x/OPEN* Portability Guide 4.0.

The *MAILER* environment variable is an extension to traditional implementations of **calendar**.

SEE ALSO

mailx(1)

MPE/iX NOTES

The current MPE/iX implementation of **calendar** does not allow the use of the **-** option. MPE/iX has no way of interactively obtaining a list of the home directories (home groups) of all users on the system. To determine a user's home group, use the MPE/iX CI command **LISTUSER**

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

NAME

callci — run a MPE/iX CI command from the MPE/iX Shell

SYNOPSIS

```
callci command_string
callci -
```

DESCRIPTION

The **callci** command allows you to run a MPE/iX Command Interpreter (MPE/iX CI) command while in the MPE/iX Shell. It concatenates all arguments specified on the command line together with appropriate spaces and then uses the **HPCICOMMAND** intrinsic to submit the resulting string to the MPE/iX CI for execution. If the only argument specified is **-**, **callci** reads from the standard input one line at a time and submits each line to the MPE/iX CI for execution.

EXAMPLES

The following command lists all of the files in `pub.sys`.

```
shell> callci listfile @.pub.sys
```

This next example uses the MPE/iX CI **COPY** command to do a forced copy of a file `foo` to a file `bar`. Note that the semicolon (`:`) must be quoted to prevent the MPE/iX Shell from interpreting it.

```
shell> callci copy foo,bar;yes
```

You could also enter the command as

```
shell> callci "copy foo,bar;yes"
```

DIAGNOSTICS

Possible exit status values are:

- 0 The call to MPE/iX CI was successful
- 1 **callci** was unable to invoke MPE/iX CI or a command error occurred.
- 2 Some other error occurred.

Messages

This utility generates no error messages itself; however, the MPE/iX CI or the command being executed may generate messages.

PORTABILITY

This command is unique to MPE/iX Shell and Utilities.

MPE/iX NOTES

`callci` is available as both a built-in shell utility and an external utility.

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

NAME

cat — concatenate and display text files

SYNOPSIS

cat [-su] [-v[et]] [*file* ...]

DESCRIPTION

cat displays and concatenates files. It copies each *file* argument to the standard output. If you specify no files or give - as a file name, **cat** reads the standard input.

Options

cat accepts the following options:

- e displays a \$ character at the end of each line. This option only works if you also specify -v.
- s does not produce an error message if **cat** cannot find or read a specified file.
- t displays tabs as ^I. This option only works if you also specify -v.
- u does not buffer output.
- v displays all characters including those that are unprintable. If the most significant bit in a character is set and the character is non-printable, **cat** displays the character without that bit, but precedes it with the characters M-. **cat** displays other unprintable characters as ^ followed by the character representing the control character (for example, ^A for CTRL-A).

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

Message: *filename* : system error

Cause: See **syserror**(3).

Action: See **syserror**(3).

Message: input file "*filename*" is identical with output

Cause: You specified *filename* as both an input and output file. It is also possible that the output file was linked to *filename*.

Action: Use a file other than *filename* as the output file.

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **cat**.
Action: Check the *DESCRIPTION* section for a list of valid **cat** options.

Message: write error on standard output: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

The **-e**, **-s**, **-t**, and **-v** options are extensions to the POSIX standard.

SEE ALSO

cp(1), **more(1)**, **mv(1)**

MPE/iX NOTES

On MPE/iX, **cat** is available as both a built-in shell utility and an external utility.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

NAME

cd — change working directory

SYNOPSIS

cd [*directory*]

cd *old new*

DESCRIPTION

The command

cd *directory*

changes the working directory of the current shell execution environment (see **sh(1)**) to *directory*. If *directory* is an absolute path name, beginning with /, this is the target directory. If *directory* is a relative path name, **cd** assumes it is relative to the current working directory.

In the shell, if the variable *CDPATH* is defined, the built-in **cd** command searches for a relative path name under each of the directories defined in *CDPATH*. If **cd** finds the directory outside the current working directory, it displays the new working directory.

Use colons to separate directories in *CDPATH*. In *CDPATH*, a null string represents the current directory. For example, if the value of *CDPATH* begins with a separator character, **cd** searches the current directory first; if it ends with a separator character, **cd** searches the current directory last.

In the shell, the command

cd -

is a special case that changes the working directory to the previous working directory by exchanging the values of the variables *PWD* and *OLDPWD*.

Note: Repeating this command toggles the current working directory between the current and old working directory.

Calling **cd** without arguments sets the working directory to the value of the *HOME* environment variable, if the variable exists. If there is no *HOME* variable, **cd** does not change the current directory.

The form

cd *old new*

is an extension to traditional KornShell implementations. The shell keeps the name of the current directory in the variable *PWD*. The **cd** command scans the current value of *PWD* and replaces the first occurrence of the string *old* with the string *new*. The shell displays the resulting value of *PWD* and it becomes the new working directory.

If either *directory* is a symbolic link to another directory, the behavior depends upon the setting of the shell's `-o logical` flag. See `set(1)` for more information.

ENVIRONMENT VARIABLES

CDPATH

contains a list of directories for `cd` to search under when *directory* is a relative path name.

HOME

contains the name of your home directory. This is used when you do not specify *directory* on the command line.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - no *HOME* directory
 - no previous directory
 - a search for *directory* failed
 - an *old/new* substitution failed
- 2 An invalid command line option.

Messages

Message: `"dir" bad directory: system error`

Cause: See `syserror(3)`.

Action: See `syserror(3)`.

Message: `cd: restricted`

Cause: You were using the restricted version of the shell (for example, by specifying the `-r` option for `sh`). The restricted shell does not allow the `cd` command.

Action: To use the `cd` command, you must be using a non-restricted shell.

Message: `no HOME directory`

Cause: You attempted to use `cd` to return to your home directory; however, the environment variable *HOME* was not defined.

Action: Set the environment variable *HOME* to the path name of your home directory.

Message: `no previous directory`

Cause: You tried the command `cd -` to return to your previous directory; however, there was no record of what your previous directory was.

Action: Make sure that you only use `cd -` when there is a previous directory.

Message: pattern "*old*" not found in "*dir*"

Cause: You tried a command of the form

```
cd old new
```

However, the name of the current directory *dir* does not contain any string matching the regular expression *old*.

Action: Ensure that the name of the current directory contains the regular expression *old*.

Message: Unknown option "*-option*"

Cause: You specified an option that is not valid for **cd**

Action: Check the *DESCRIPTION* section for a list of valid **cd** options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

All UNIX systems feature the first form of the command.

In the MPE/iX Shell and Utilities implementation of this command, all forms are built into the shell.

The **cd *old new*** form of the command is an extension to the POSIX standard.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

set(1) **sh(1)**

NAME

chgrp — change the group ownership of files and/or directories

SYNOPSIS

chgrp [-fR] *group pathname ...*

DESCRIPTION

chgrp sets the group ID to *group* for the files and directories named by the *pathname* arguments. *group* may be a group name from the group database, or a numeric group ID.

Options

chgrp accepts the following options:

- f does not issue an error message if **chgrp** cannot change the group ID. In this case, **chgrp** always returns a status of zero.
- R If a *pathname* on the command line is the name of a directory, **chgrp** changes the group ID of all files and subdirectories under that directory. If **chgrp** cannot change some file or subdirectory under the directory, it continues to try to change the other files and subdirectories under the directory, but exits with a non-zero status.

DIAGNOSTICS

Possible exit status values are:

- 0 You specified -f, or **chgrp** successfully changed the group ownership of all the specified files and directories.
- 1 Failure due to any of the following:
 - unable to access a specified file
 - unable to change the group of a specified file
 - encountered a fatal error when you specified -R option
- 2 Failure due to any of the following:
 - the command line contained an unknown option or too few arguments
 - **chgrp** did not recognize the specified *group*

Messages

Message: cannot determine OPEN_MAX: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

-
- Message:** fatal error during "-R" option
Cause: You specified the **-R** option but some file or directory in the directory structure was inaccessible.
Action: Make sure that you have access to all files in the directory structure.
- Message:** file "*filename*" : *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** file "*filename*" : You are not a member of the *group* group
Cause: You attempted to change the group ownership of *filename* to *group*, but you are not a member of the specified group.
Action: Specify a group to which you belong.
- Message:** group "*group*" is unknown
Cause: You specified a group name that **chgrp** was unable to find in the group database.
Action: Specify a valid group name or use a valid numeric group ID.
- Message:** read directory "*pathname*" : *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** stat file "*filename*" : *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **chgrp**.
Action: Check the *Options* section of this man page for a list of valid options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

The **-f** option is an extension to the POSIX standard.

MPE/iX NOTES

The current MPE/iX implementation of **chgrp** does not allow you to modify the group ID of the root directory, account directories, and group directories.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

chmod(1), **chown(1)**

NAME

chmod — change access permissions of a file

SYNOPSIS

chmod [-fR] *mode pathname*

DESCRIPTION

chmod changes the access permissions or *modes* of the specified files or directories. Modes determine who can read, change or execute a file.

Options

chmod accepts the following options:

- f Does not issue error messages concerning file access permissions, even if **chmod** encounters such errors.
- R If you specify a directory as a path name on the command, **chmod** changes the access permissions of all files and subdirectories under that directory.

Modes

You can specify the *mode* value on the command line in either symbolic form or as an octal value.

A symbolic *mode* has the form

[*who*] *op permission* [*op permission* . . .]

The *who* value is any combination of the following:

- u Sets user (individual) permissions.
- g Sets group permissions.
- o Sets other permissions.
- a Sets all permissions; this is the default.

The *op* part of a symbolic mode is an operator that tells **chmod** to turn the permissions on or off. The possible values are:

- + Turns on a permission.
- Turns off a permission.
- = Turns on the specified permissions and turns off all others.

The *permission* part of a symbolic mode is any combination of the following:

- r Read permission. If this is off, you cannot read the file.

-
- x Execute permission. If this is off, you cannot execute the file.
 - X Execute/search permission for a directory; or execute permission for a file only when the current mode has at least one of the execute bits set.
 - w Write permission. If this is off, you cannot write to the file.
 - s On POSIX-compliant and UNIX systems, this stands for *setuid on execution* or *setgid on execution* permission.
 - t On UNIX systems, this stands for the *sticky* bit.

You can specify multiple symbolic modes if you separate them with commas.

Absolute modes are octal numbers specifying the complete list of attributes for the files; you specify attributes by OR'ing together these bits.

4000	Setuid bit
2000	Setgid bit
1000	Sticky bit
0400	Individual read
0200	Individual write
0100	Individual execute (or list directory)
0040	Group read
0020	Group write
0010	Group execute
0004	Other read
0002	Other write
0001	Other execute

EXAMPLES

```
chmod -w nowrite
```

makes file `nowrite` read-only.

```
chmod a=rwx file
```

turns on read, write, and execute permissions, and turns off the `setuid` and `sticky` bit attributes. This is equivalent to

```
chmod 0777 file
```

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - unable to access a specified file
 - unable to change the modes on a specified file
 - unable to read the directory containing item to change
 - encountered a fatal error when using the **-R** option
- 2 Failure due to any of the following:
 - missing or invalid *mode* argument
 - too few arguments.

Messages

- Message:** fatal error during "-R" option
Cause: You specified the **-R** option but some file or directory in the directory structure was inaccessible.
Action: Check all files in the directory structure to make sure that you have access to them.
- Message:** file "*filename*": system error
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Insufficient memory
Cause: There were not enough free system resources to perform the specified operation.
Action: Free up more resources.
- Message:** Missing mode argument.
Cause: You did not specify an argument represent the new access permissions.
Action: Provide the missing argument.
- Message:** read directory "*pathname*": system error
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** stat file "*filename*": system error
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

-
- Message:** Unknown or missing operator in symbolic mode "*modestring*"
- Cause:** When using the symbolic mode to indicate new access permissions, you specified a string *modestring* which was either missing an operator or contained an operator that **chmod** does not recognize.
- Action:** Make sure that all *mode* values in symbolic mode contain one of the following operators: +, -, or =.
- Message:** Octal mode may contain only digits [0-7] in *numstring*
- Cause:** When using the octal mode to indicate new access permissions, you specified a string *numstring* which contained a character other than the digits 0 to 7.
- Action:** Make sure that all *mode* values in octal mode are valid octal numbers, containing only the digits 0 through 7.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

The **-f** option and the **t** permission are extensions to the POSIX standard.

MPE/iX NOTES

The current MPE/iX implementation of **chmod** does not allow you to modify the access permissions of the root directory, account directories, and group directories. Also, use of the *sticky* bits is not allowed.

On MPE/iX, **chmod** is available as both a built-in shell utility and an external utility.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

lc(1), **ls(1)**, **umask(1)**

NAME

chown — change the ownership of files and/or directories

SYNOPSIS

chown [-**fR**] *owner* [:*group*] *pathname* ...

DESCRIPTION

chown sets the user ID to *owner* for the files and directories named by *pathname* arguments. *owner* can be a user name from the user database the user database, or a numeric user ID.

If you include a *group* name (specify the *owner* followed immediately by a colon (:) and *group* with no intervening spaces, such as *owner:group*) then **chown** also sets the group ID to *group* for the files and directories named.

Options

chown accepts the following options:

- f** does not issue an error message if **chown** cannot change the owner. In this case, **chown** always returns a status of zero.
- R** If a *pathname* on the command line is the name of a directory, **chown** changes all the files and subdirectories under that directory to belong to the specified *owner* (and *group*, if *:group* is specified). If **chown** cannot change some file or subdirectory under the directory, it continues to try to change the other files and subdirectories under the directory, but exits with a non-zero status.

DIAGNOSTICS

Possible exit status values are:

- 0 You specified **-f**, or **chown** successfully changed the ownership of all the specified files and directories.
- 1 Failure due to any of the following:
 - unable to access a specified file
 - unable to change the owner of a specified file
 - unable to read the directory containing the directory entry of the file
 - encountered a fatal error when using the **-R** option
- 2 Failure due to any of the following:
 - the command line contained an invalid option
 - the command line had too few arguments
 - specified an *owner* with a userid that the system did not recognize

Messages

- Message:** fatal error during "-R" option
Cause: You specified the **-R** option but some file or directory in the directory structure was inaccessible.
Action: Check all files in the directory structure to make sure that you have access to them.
- Message:** file "*filename*" : *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** group "*group*" is unknown
Cause: You specified a group name that **chown** was unable to find in the group database.
Action: Specify a valid group name or use a valid numeric group ID.
- Message:** read directory "*pathname*" : *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** stat file "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **chown**.
Action: Check the *Options* section of this man page for a list of valid options.
- Message:** user "*user*" is unknown
Cause: You specified a user name that **chown** was unable to find in the user database.
Action: Specify a valid group name or use a valid numeric group ID.

PORTABILITY

POSIX.2. *x/OPEN* Portability Guide 4.0. All UNIX systems.

The **-f** option is an extension to the POSIX standard.

MPE/iX NOTES

The current MPE/iX implementation of **chown** does not allow you to modify the ownership of the root directory, account directories, and group directories.

On MPE/iX, **chown** is available as both a built-in shell utility and an external utility.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

chown(1)

MPE/iX Shell and Utilities

chown(1)

SEE ALSO

chgrp(1), chmod(1)

NAME

ci — check in a file under RCS

SYNOPSIS

```
ci [-B] [-ddate] [-Ffile...] [-f[rev]] [-G] [-g] [-h[rev]] [-I] [-k[rev]]
[-l[rev]] [-mmsg] [-Nname] [-nname] [-O] [-q[rev]] [-Rdiff_exec]
[-r[rev]] [-sstate] [-T] [-t[txtfile]] [-u[rev]] [-wlogin] [-Yfile] file ...
```

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

ci adds a new revision of a working file to the corresponding RCS file. Executing this command on a file is referred to as *checking in* the file.

For **ci** to work, the user's login name must be on the access list, except if the access list is empty or the user is the system administrator or the owner of the file. To append a new revision to an existing branch, the tip revision on that branch must be locked by the user; otherwise, a new branch must be created. This restriction is not enforced for the owner of the file, unless locking is set to *strict* (see **racs**(1)). A lock held by someone else may be broken with the **racs** command.

Normally, **ci** checks whether the revision to be checked in is different from the preceding one. If it is not different, **ci** either aborts the check-in (if **-q** is given) or asks whether to abort (if **-q** is omitted). A check-in can be forced with the **-f** option.

For each revision checked in, **ci** prompts for a log message. The log message should summarize the change and must be terminated with a line consisting of a single dot (.), the string ~., or an end of file. Log messages may be edited using the facilities described in **rcsed**(3).

If several files are checked in, **ci** asks whether to reuse the previous log message. If the standard input is not a terminal, **ci** suppresses the prompt and uses the same log message for all files. See also **-m**.

The number of the checked-in revision can be given by any of the options **-r**, **-f**, **-k**, **-l**, **-u**, **-h**, or **-q**.

If the RCS file does not exist, **ci** creates it and checks in the contents of the working file as the initial revision. The default number for this revision is 1.1, and the access list is initialized to empty. Instead of the log message, **ci** requests descriptive text (see **-t**). This descriptive text may be edited using the facilities described in **rcsed**(3). If you specify the **-r** option, you are asked for both descriptive text and a log message.

If the configuration file specifies *Compress*, **ci** compresses the RCS file after the revision has been checked in. It is not necessary to specify anything about compression when you check it

out again because all RCS commands automatically check to see if a file is compressed before performing an operation on them. The checked-out working file is not compressed.

Note: If an RCS file is in the midst of a check-in operation, it cannot be accessed by any RCS command.

Options

ci accepts the following options:

- B** forces **ci** to check in the revision using binary format. If the revision is actually a text file, RCS still produces correct results, but may work less efficiently than it would if text format was used. **-B** cannot be specified if the RCS file already contains revisions checked in using text format or the RCS file was created with the **rcs -T** command.
- d***date* uses *date* for the check-in date and time. The *date* option may be specified in free format as explained in **co**(1). This option is useful for altering the check-in date, and for **-k** if no date is available.
- F***file...* provides an alternate way to specify file names. The given *file* is a text file containing a list of file names, one file name per line. **ci** checks in all the files named in *file*, using the options specified on the command line. Multiple **-F** options may be specified on the command line, and can either be grouped together or interspersed between options.
- f**[*rev*] forces a check in. The new revision is checked in even if it is not different from the preceding one.
- G** sets the RCS file date to the current date. Normally, when updating, the date stamp of the RCS file is set to the check-in date of the head revision.
- g** sets the check-in date of the revision to the modification date of the working file. Normally, **ci** uses the current time as the check-in date.
- h**[*rev*] uses the **-h** (half-hearted) option to **diff**(1) to produce the set of changes for this revision. Normally, this half-hearted algorithm is used only if **diff** runs out of system resources using the normal algorithm.
- I** allows **ci** to accept redirected input from a file or from a pipe instead of standard input. Input is a sequence of strings separated by lines containing only a single dot.
- k**[*rev*] searches the working file for keyword values to determine its revision number, creation date, state, and author (see **co**(1)). **ci** then assigns these values to the

checked-in revision, rather than computing them locally. It also generates a default log message noting the login name of the user and the actual check-in date. This option is useful for software distribution. A revision that is sent to several sites should be checked in with the **-k** option at these sites to preserve the original revision number, date, author, and state. The extracted keyword values and the default log message may be overridden with the options **-r**, **-d**, **-s**, **-w**, and **-m**.

-l[rev]

checks in a new revision of a working file to the corresponding RCS file, while leaving a locked copy of the working file intact. This is useful for saving a revision when you want to continue editing it after the check-in.

-mmsg uses the string *msg* as the log message for all revisions checked in.

-Nname

same as **-n**, except that it overrides a previous assignment of *name*.

-nname

assigns the symbolic name *name* to the number of the checked-in revision. **ci** prints an error message if *name* is already assigned to another revision.

-O ignores any default options found in the local configuration file.

-q[rev]

quiet mode; diagnostic output is not printed. If there is no difference between the revision being checked in and the previous revision of the file, the new revision is not checked in unless **-f** is specified.

-Rdiff_exec

specifies the path name of the **diff** command to use. This allows you to use your own system **diff** for everything but RCS.

-r[rev]

assigns the revision number *rev* to the checked-in revision, releases the corresponding lock, and deletes the working file. This is the default. *rev* may be symbolic or numeric.

If *rev* is a revision number, it must be higher than the latest one on the branch to which *rev* belongs, or must start a new branch.

If *rev* is a branch rather than a revision number, the new revision is appended to that branch. The level number is obtained by incrementing the tip revision number of that branch. If *rev* indicates a non-existing branch, that branch is created with the initial revision numbered *rev.1*. You can have a mixture of numeric and symbolic parts when you do this.

If *rev* is omitted, **ci** tries to derive the new revision number from the user's last lock.

If the user has locked the tip revision of a branch, the new revision is appended to that branch. The new revision number is obtained by incrementing the tip revision number. If the user locked a non-tip revision, **ci** starts a new branch at that revision by incrementing the highest branch number at that revision. The default initial branch and level numbers are 1.

If you do not specify *rev* and you have no lock, but you are the owner of the file and locking is not set to *strict*, the revision is appended to the default branch (normally the trunk; see **rcs -b**).

-sstate

sets the state of the checked-in revision to the identifier *state*. The default state is *Exp*.

-T

forces **ci** to check in the revision using text format. If the revision is actually a non-text file, RCS may not work properly. **-T** cannot be specified if the RCS file already contains revisions checked in using binary format or the RCS file was created with the **rcs -B** command.

-t[txtfile]

writes descriptive text into the RCS file (deleting the existing descriptive text). If *txtfile* is omitted, **ci** prompts you for text supplied from the standard input. You may enter any number of lines of text. When you have finished entering the description, enter a line containing a single dot (*.*), the string *~.*, or the end-of-file character. **rcsed**(3) describes facilities for editing the descriptive text as you enter it.

If *txtfile* is specified, the descriptive text is copied from that file. During the initial check-in of a file, descriptive text is requested even if **-t** is not given. The prompt is suppressed if standard input is not a terminal.

-u[rev]

works like **-l**, except that the checked-in revision is not locked. This is useful if you want to process (for example, compile) the revision immediately after check-in.

-wlogin

uses *login* for the author field of the checked in revision. This is useful for altering the author, and for **-k** if no author is available.

-Yfile specifies *file* as the local configuration file.

File Modes

An RCS file created by **ci** inherits the read and execute permissions from the working file. If the RCS file exists already, **ci** preserves its read and execute permissions. **ci** normally turns off all write permissions of RCS files.

Files

The person who enters the command must have read/write permission for the directories containing the RCS file and the working file, and read permission for the RCS file itself. A number of temporary files are created. A semaphore file is created in the directory containing the RCS file. **ci** always creates a new RCS file and unlinks the old one. This strategy makes links to RCS files useless.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred while checking in one of the specified *files*.

For each revision, **ci** prints the RCS file, the working file, and the number of both the checked-in and the preceding revision (unless you use **-q** to request quiet mode).

Messages

Message: `-d` overrides `-g`

Cause: You specified both the `-d` option and the `-g` option. These two options are mutually exclusive. The command continues as though only the `-d` option was specified.

Action: If you do not want the behaviour described under **CAUSE**, specify only one of the two options.

Message: Branch point does not exist for *rev*

Cause: You attempted to check in an initial revision of a file on a branch.

Action: The first revision checked in must be on the main trunk.

Message: Can't adjust mode of *filename* See **syserror(3)**.

Action: See **syserror(3)**.

Message: Can't expand file *filename*: *system error*

Cause: See **syserror(3)**.

Action: See **syserror(3)**.

Message: Can't expand keywords in *workfile*: *system error*

Cause: See **syserror(3)**.

Action: See **syserror(3)**.

Message: Can't find a date in *workfile*

Cause: You attempted to check in *workfile* using the `-k` option, but *workfile* does not contain any RCS keywords that provide the date.

Action: Specify the desired date the command line using the `-d` option.

-
- Message:** Can't find a revision number in *workfile*
Cause: You attempted to check in *workfile* using the **-k** option, but *workfile* does not contain any RCS keywords that provide the revision number.
Action: Specify the desired revision number on the command line using the **-r** option.
- Message:** Can't find a state in *workfile*
Cause: You attempted to check in *workfile* using the **-k** option, but *workfile* does not contain any RCS keywords that provide the state.
Action: Specify the desired state on the command line using the **-s** option.
- Message:** Can't find an author in *workfile* You attempted to check in *workfile* using the **-k** option, but *workfile* does not contain any RCS keywords that provide the author.
Action: Specify the desired author on the command line using the **-w** option.
- Message:** Can't open file *textfile* with description: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Can't parse date/time: *dates*
Cause: You specified an improperly formatted date/time string with the **-d** option.
Action: Check the *DESCRIPTION* section for the proper format of the date/time string.
- Message:** Can't set mode of *filename*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Can't reread redirected stdin for log message; use **-m**
Cause: When running **ci** non-interactively, you did not specify a log message with the **-m** option and did not use the **-I** option to read the log message from the standard input.
Action: When running **ci** non-interactively, specify the **-I** option to read the log message from standard input or use the **-m** option to specify the message.
- Message:** Can't take both log and description from redirected stdin; use **-ttextfile**
Cause: You specified the **-t** option without providing the name of the text file from which the new descriptive text should be taken. **ci** cannot read both the new descriptive text and the revision log message from the terminal.
Action: Place the new descriptive text in a text file and run **ci** again, using the **-t** option to specify that file.
- Message:** Can't write new RCS file *name1*; saved in *name2*: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

-
- Message:** cannot change file format of existing RCS file *filename*
Cause: You attempted to check a text file in as a binary file, or vice versa, and revisions already existed in the other format.
Action: Check the file in using the existing format.
- Message:** Cannot find branchpoint *rev*
Cause: You attempted to check in a branch revision for which there is no branch point.
Action: Make sure that the revision number that you are attempting to create is correct. If it is, then create the branch revision necessary.
- Message:** checkin aborted since *workfile* was not changed.
Cause: During a non-interactive check-in, **ci** determined that the workfile was identical to the revision on which it was based. Under these circumstances, does not check in the file.
Action: If you want the check-in to proceed regardless, specify the **-f** option.
- Message:** Date *date1* is earlier than *date* in existing revision *rev*
Cause: You attempted to check in a file with a date that was earlier the date associated with the previous revision on the same branch.
Action: Make sure that the date being used to check in the file is more recent than that of the previous revision.
- Message:** deltanumber *rev* too low; must be higher than *head_rev*
Cause: You specified a revision number *rev* that was lower than *head_rev*, the head of the current branch.
Action: Make sure that the revision number you typed in was correct. If it was, then check your revision in as a branch from the main trunk.
- Message:** Error closing input file *filename*
Cause: The RCS file *filename* is compressed and an error occurred during decompression.
Action: The file is possibly corrupt. Try again with another copy of the file.
- Message:** Inconsistent date/time: *date*
Cause: You specified an argument with the **-d** option that was not a valid date and time.
Action: Make sure that the time you entered was correct, and try again.
- Message:** line too long -- truncated
Cause: You specified a message as an argument to the **-m** that was longer than the maximum length permitted. The message was truncated to this length.
Action: Specify a shorter message.
- Message:** Missing author for **-w** option
Cause: You specified the **-w** option, but did not provide an author as its argument.
Action: Provide the missing author.

-
- Message:** Missing date for `-d` option
Cause: You specified the `-d` option, but did not provide a date as its argument.
Action: Provide the missing date.
- Message:** Missing message for `-m` option
Cause: You specified the `-m` option, but did not provide a log message as its argument.
Action: Provide the missing log message.
- Message:** Missing name for `-option` option
Cause: You specified the `-n` or `-N` option without providing a symbolic name on the command line or in the configuration file.
Action: Provide the missing symbolic name.
- Message:** Missing state for `-s` option
Cause: You specified the `-s` option, but did not provide a state as its argument.
Action: Provide the missing state.
- Message:** No filename present for `-F` option.
Cause: You specified the `-F` option, but did not provide a file name as its argument.
Action: Provide the missing file name.
- Message:** No input file.
Cause: You failed to specify an input file on the command line.
Action: Provide the missing file name.
- Message:** no lock set by *username*
Cause: You attempted to check in a file for which you had no locks.
Action: Lock a revision using the `-l` option of `rcs`.
- Message:** no lock set by *username* for revision *rev*
Cause: You attempted to check in a revision that was not locked.
Action: Lock the revision *rev* using the `-l` option of `rcs`.
- Message:** Redefinition of `-option` option
Cause: You specified the `-d`, `-m`, `-s`, `-t`, or `-w` option more than once on the command line.
Action: Specify each of the given options only once on a command line.
- Message:** Redefinition of file format specifier
Cause: You specified both the `-B` option (binary) and the `-T` option (text) on the same command line.
Action: Specify either `-B` or `-T`, but not both.

-
- Message:** Redefinition of revision number
Cause: You have specified more than one revision to be checked in. The **-f**, **-h**, **-k**, **-l**, **-q**, **-r**, and **-u** options can each take a revision as an argument; however, only one option per command line may specify a revision.
Action: Specify, at most, a single revision.
- Message:** Redefinition of symbolic name
Cause: You specified multiple **-n** or **-N** options with accompanying symbolic names.
Action: Specify, at most, one **-n** or **-N** option.
- Message:** revision *rev* locked by *username*
Cause: You attempted to check in a revision that was already locked by the user *username*.
Action: Either ask *username* to unlock the required revision, or break the lock using the **-u** option of **rcs**.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **ci**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **ci** options.
- Message:** working file *workfile* not readable or nonexistent
Cause: You specified an RCS file on the command line, but the corresponding work file does not exist.
Action: Check that the specified RCS file exists, and was named properly. Also check the value of the environment variable *RCSPATH*. Try to check in the work file instead. the **-t** option to identify the text file.

For a list of the error messages that may occur when editing a log message, see **rcsed**(3).

PORTABILITY

All UNIX systems.

The **-B**, **-F**, **-g**, **-G**, **-I**, **-O**, **-q**, **-R**, **-T**, and **-Y** options are extensions to traditional implementations of **ci**.

LIMITS

Compressed files, encrypted files, and files checked in using binary file format are not handled properly by traditional versions of RCS. If an RCS file is compressed and you want to manipulate it with a traditional version of RCS, you must **uncompress** the file first.

MPE/iX NOTES

The current MPE/iX implementation of **ci** has the following limitations:

- Due to the fact that the comma (,) is not a valid character in MPE/iX file names, the traditional ,v naming convention is not currently implemented. This means that you must have a subdirectory named RCS under your current directory or RCS will not work properly.
- For compatibility with traditional MPE/iX security features, you can only rename a file in an MPE/iX group if you own the file or if you have SM capability. Because this command attempts to rename files to which it may not have write access, this feature can interfere with the command's operation. For this reason, RCS archives should not be placed in an MPE/iX group, but rather should be located elsewhere in the hierarchical file system where the restriction does not apply. For example, the default *RCSPATH* places the archives in a subdirectory named RCS in the current working directory.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

co(1), **ident(1)**, **rcs(1)**, **rcsclean(1)**, **rcsdiff(1)**, **rcsmerge(1)**, **rlog(1)**,
uncompress(1), **rcsedit(3)**, **rcsfile(3)**

NAME

cksum — compute checksum and byte count for file

SYNOPSIS

cksum [-**ciprt**] [*file* ...]

DESCRIPTION

cksum calculates and displays a checksum for each input *file*. A checksum is an error-checking technique used by many programs as a quick way to compare files that have been moved from one location to another to ensure that no data has been lost. **cksum** also displays the number of eight-bit bytes in each *file*. If you do not specify any *files* or you specify - as the file name, **cksum** reads the standard input.

cksum differs from **sum** only in the format of the output. **cksum**'s output has the form

checksum bytecount filename

where *bytecount* is the number of bytes in the file.

cksum can calculate checksums in a variety of ways. The default is compatible with the POSIX.2 standard. You can specify other algorithms by specifying options. The POSIX standard does not recognize these algorithms; we provide them for compatibility with the **sum** command.

Options

cksum accepts the following options:

- c** uses a standard 16-bit Cyclical Redundancy Check (CRC-16).
- i** uses the CCITT standard Cyclic Redundancy Check (CRC-CCITT). Data communications network protocols often use a cyclic redundancy check to ensure proper transmission. This algorithm is more likely to produce a different sum for inputs — the only difference is byte order.
- p** uses the POSIX.2 checksum algorithm. This is the default.
- r** enables the use of an alternate checksum algorithm which has the advantage of being sensitive to byte order.
- t** produces a line containing the total number of bytes of data read as well as the checksum of the concatenation of the input files.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

Message: input file "*filename*": *system error*

Cause: See **syserror(3)**.

Action: See **syserror(3)**.

Message: Unknown option "*-option*"

Cause: You specified an option that is not valid for **cksum**.

Action: Check the *DESCRIPTION* section of this man page for a list of valid **cksum** options

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

All of the listed options are extensions to the POSIX standard.

SEE ALSO

cmp(1), **diff(1)**, **ls(1)**, **sum(1)**, **wc(1)**

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

NAME

cmp — compare two files

SYNOPSIS

cmp [-b1sx] file1 file2 [seek1[seek2]]

DESCRIPTION

cmp compares two files. If either file name is **-**, **cmp** reads the standard input for that file. By default, **cmp** begins the comparison with the first byte of each file. If you specify *seek1* and/or *seek2*, **cmp** uses it as a byte offset into *file1* or *file2* (respectively), and comparison begins at that offset instead of at the beginning of the files. The comparison continues (one byte at a time) until a difference is found, at which point the comparison ends and **cmp** displays the byte and line number where the difference occurred. **cmp** numbers bytes and lines beginning with 1.

Options

cmp accepts the following options:

- b** compares single blocks at a time. Normally, **cmp** reads large buffers of data into memory for comparison.
- l** causes the comparison and display to continue to the end. **cmp** displays the byte number (in decimal) and the differing bytes (in octal) for each difference found. **cmp** attempts no resynchronization.
- s** suppresses output and returns a non-zero status if the files differ.
- x** displays the differing bytes shown by the **-l** option in hex.

DIAGNOSTICS

Possible exit status values are:

- 0 The files were identical.
- 1 The files were not identical.
- 2 Failure because of an error opening or reading an input file.

Messages

Message: indecipherable seek address "*string*"
Cause: You specified an invalid seek address.
Action: Specify a valid seek address.

Message: *filename* : system error
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: too few or too many args
Cause: You specified an incorrect number of arguments on the command line.
Action: Make sure that you specify two file names and no more than two seek addresses on the command line.

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **cmp**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **cmp** options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

The **-b** and **-x** options and the seek pointers are extensions to the POSIX standard.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

comm(1), **diff(1)**, **uniq(1)**

NAME

co — check out a file under RCS

SYNOPSIS

```
co [-ddate] [-f[rev]] [-Ffile...] [-G] [-g] [-h] [-I] [-jjoinlist] [-K] [-k]
[-l[rev]] [-O] [-p[rev]] [-q[rev]] [-r[rev]] [-sstate] [-u[rev]] [-wauthor]]
[-x] [-Yfile] file ...
```

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

co checks out the most recent revision of a file from the corresponding RCS file and makes it available in a working file.

Revisions from an RCS file may be checked out *locked* or *unlocked*. Locking a revision prevents overlapping updates. A revision checked out for reading or processing (that is, compiling) need not be locked. A revision checked out for editing should usually be locked, to prevent someone else from editing the file at the same time. If you try to check out a revision and lock it, the operation fails if the revision is already locked by another user. (A lock may be broken with the **racs(1)** command if you insist.)

To use co with locking, you must be on the access list of the RCS file, unless you are the owner of the file, you are the system administrator, or the access list is empty. co without locking is not subject to access list restrictions, and is not affected by the presence of locks.

Revisions may be selected for check-out according to revision number, branch number, check-in date/time, author, or state. This is done by putting options after the co command. When two or more options are specified in combination, co retrieves the latest revision that satisfies all of them. If no options are specified, co retrieves the latest revision on the default branch (normally the trunk; see the -b option of **racs(1)**). A revision or branch number may be attached to any of the options -f, -l, -p, -q, -r, or -u. The options -ddate, -sstate, and -wauthor retrieve a revision with particular identification fields from the default branch or the branch specified by one of -f, -l, -p, -q, -r, or -u.

A co command applied to an RCS file with no revisions creates a zero-length working file.

co suppresses keyword expansion unless the configuration file specifies KeywordExpand. If you specify the -k or -K options, keyword expansion is suppressed in any case.

You do not need to specify whether the RCS file is compressed when checking out a revision. For all commands, RCS automatically determines whether or not the file is compressed; in either case, the checked-out working file is not compressed.

Options

co accepts the following options:

-d*date* retrieves the latest revision on the selected branch with a check-in date/time that is less than or equal to *date*. The date and time may be given in free format and are converted to local time. Below we give some examples of formats for *date*:

```
22-April-1982, 17:20-CDT
2:25 AM, Dec. 29, 1983
Tue-PDT, 1981, 4pm Jul 21
Fri, April 16 15:52:25 EST 1982
```

If parts of the date or time are omitted, **co** determines the defaults in the order year, month, day, hour, minute, and second (most to least significant). At least one of the fields of the date/time must be provided. For omitted fields that are of higher significance than the highest provided field, the current values are assumed. For all other omitted fields, the lowest possible values are assumed. For example, the date 20 , defaults to 10:30:00 of the 20th of the current month and current year. The date/time must be enclosed in quotes if it contains spaces.

-f[*rev*]

forces the overwriting of the working file. This is useful in connection with **-q**. Also see the *File Modes* subsection.

-F*file* provides an alternate way to specify file names. The given *file* is a text file containing a list of file names, one file name per line. **co** checks out all the files named in *file*, using the options specified on the command line. Multiple **-F** options may be specified on the command line, and can either be grouped together or interspersed between options.

-G sets the RCS file date to the current date. Normally, when updating, the date stamp of the RCS file is set to the check-in date of the head revision.

-g sets the date of the working file to the current date. Normally, it is set to the check-in date of the revision.

-h forces **diff** to use the **-h** option when called by **co**. This saves time if you know that the file is large enough to require the **-h** option.

-I allows **co** to accept redirected input from a file or input from a pipe instead of standard input. Input is a sequence of strings separated by lines containing only a single dot.

-j*joinlist*

is typically used to take changes from one branch and incorporate them into another branch. Technically speaking, **-j** generates a new revision which is the *join* of the revisions on *joinlist*. A join operation works with three revisions: *root*, *rev1*, and

rev2. **co** determines all the changes needed to change *root* into *rev1* and applies those changes to a copy of *rev2*. This is particularly useful if *rev1* and *rev2* are the ends of two branches that have *root* as a common ancestor.

If $rev1 < root < rev2$ on the same branch, joining generates a new revision which is like *rev2* but with all changes that lead from *rev1* to *root* undone. If changes from *root* to *rev1* overlap with changes from *root* to *rev2*, **co** prints a warning and includes the overlapping sections, delimited by the lines:

```

<<<<<<<  rev1
lines in rev1
=====
lines in rev2
>>>>>>>  rev2

```

In the option **-j***joinlist*, *joinlist* is a comma-separated list of colon-separated pairs, of the form *root:rev2*, where *root* and *rev2* are (symbolic or numeric) revision numbers.

For the initial pair in the *joinlist*, *rev1* is taken to be the revision selected by previous options to the **co** command, such as **-r**, **-l**, etc. For later pairs in the list, *rev1* is the revision generated by the previous pair. The output of one join becomes the input to the next.

For the initial pair in the list, *root* may be omitted. The default is the latest common ancestor of *rev1* and *rev2*. If any of the arguments indicate branches, the latest revisions on those branches are assumed. The options **-l** and **-u** lock or unlock *rev1*.

- K** suppresses keyword expansion and removes all existing expansions. This option takes precedence over the **-k** option.
- k** suppresses keyword expansion, even if `KeywordExpand` is specified in the configuration file. This is useful for non-readable binary files such as object files.
- l**[*rev*] same as **-r**, except that it also locks the retrieved revision for the person checking out the file. See **-r** for handling of the revision number *rev*.
- O** ignores any default options found in the local configuration file.
- p**[*rev*] prints the retrieved revision on the standard output rather than storing it in the working file. This option is useful when **co** is part of a pipeline.
- q**[*rev*] quiet mode; diagnostics are not printed.

-r[*rev*]

retrieves the latest revision number that is less than or equal to *rev*. If *rev* indicates a branch rather than a revision, the latest revision on that branch is retrieved. If *rev* is omitted, the latest revision on the default branch is retrieved (see the **-b** option for **r_{cs}(1)**). *rev* is composed of one or more numeric or symbolic names separated by a dot (.). The numeric equivalent of a symbolic field is specified with the **-n** option of the commands **ci(1)** and **r_{cs}(1)**.

-s*state*

retrieves the latest revision on the selected branch with a state that is set to *state*.

-u[*rev*]

same as **-r**, except that it unlocks the retrieved revision (if it was locked by the person checking it out). If *rev* is omitted, **-u** retrieves the latest revision locked by the user; if no such lock exists, it retrieves the latest revision on the default branch.

-w[*author*]

retrieves the latest revision on the selected branch which was checked in by the user with login name *author*. If the argument *author* is omitted, the caller's login name is assumed.

-x forces keyword expansion. This option takes precedence over the **-K** option.

-Yfile specifies *file* as the local configuration file.

Keyword Substitution

Strings of the form **\$keyword\$** and **\$keyword:...\$** embedded in the text are replaced with strings of the form **\$keyword: value\$** where *keyword* and *value* are pairs from the *Table of Keywords* shown in this man page. Keywords may be embedded in literal strings or comments to identify a revision. To use keyword expansion, you enter strings of the form

\$keyword\$

in the original file. On check-out, **co** replaces these strings with strings of the form

\$keyword: value\$

If a revision containing strings of the latter form is checked back in, the value fields is replaced during the next check-out. Thus, the keyword values are automatically updated on check-out.

Keyword expansion can be performed in all kinds of files, including binary files. If keyword expansion is inappropriate for your file, you can specify the **-k** option of **co** or add **NoKeywordExpand** to your configuration file. If you wish to remove all previous keyword expansions, specify the **-K** option of **co**.

Keyword	Meaning
\$Author\$	The login name of the user who checked in the revision.
\$Date\$	The date and time the revision was checked in.
\$Header\$	A standard header containing the full path name of the RCS file, the revision number, the date, the author, and the state.
\$Id\$	Same as \$Header\$, except that the RCS file name is without a path.
\$Locker\$	The login name of the user who locked the revision (empty if not locked).
\$Log\$	The log message supplied during check-in, preceded by a header containing the RCS file name, the revision number, the author, and the date. Existing log messages are <i>not</i> replaced. Instead, the new log message is inserted after \$Log: . . .\$. This is useful for accumulating a complete change log in a source file.
\$Name\$	The symbolic name of the revision.
\$RCSfile\$	The name of the RCS file without path.
\$Revision\$	The revision number assigned to the revision.
\$Source\$	The full path name of the RCS file.
\$State\$	The state assigned to the revision.

Table 1-4: Table of RCS Keywords

File Modes

The working file inherits the read and execute permissions of the RCS file. In addition, the owner's write permission is turned on unless the file is checked out unlocked and locking is set to *strict* (see **rcs(1)**).

If a file with the name of the working file already exists and has write permission, **co** aborts the check-out if **-q** is given, or asks whether to abort if **-q** is not given. If the existing working file is not writable or **-f** is given, the working file is deleted without asking.

Files

The person who enters the command must have write permission in the working directory, read permission for the RCS file, and either read permission (for reading) or read/write permission (for locking) in the directory which contains the RCS file.

A number of temporary files are created. A semaphore file is created in the directory of the RCS file to prevent simultaneous update.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred while checking out one of the specified files.

The RCS file name, the working file name, and the revision number retrieved are written to the diagnostic output.

Messages

- Message:** `-K` has precedence over `-k`
Cause: You specified both the `-K` option and the `-k` option. These two options are mutually exclusive. The command behaves as though only the `-K` option was specified.
Action: If you do not want the behaviour described under **CAUSE**, specify only one of the two options.
- Message:** `-l` has precedence over `-u`
Cause: You specified both the `-l` option and the `-u` option. These two options are mutually exclusive. The command behaves as though only the `-l` option was specified.
Action: If you do not want the behaviour described under **CAUSE**, specify only one of the two options.
- Message:** `-x` has precedence over `-k`
Cause: You specified both the `-x` option and the `-k` option. These two options are mutually exclusive. The command behaves as though only the `-x` option was specified.
Action: If you do not want the behaviour described under **CAUSE**, specify only one of the two options.
- Message:** Can't adjust mode of *filename*: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** cannot create "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** Can't create *workfile*; see *new_workfile*: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

-
- Message:** Can't parse date/time: *dates*
Cause: You specified a date/time string with the **-d** option that was not properly formatted.
Action: Check the *DESCRIPTION* section of this man page for the proper format for the date/time string provided with the **-d** option.
- Message:** Can't preserve mode of *filename*: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** Can't rewrite *rcsfile*; saved in: *new_rcsfile*: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** Can't unlink *filename*: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** empty join
Cause: You specified the **-j** option without providing any arguments.
Action: Provide the list of revisions that you want to join as an argument to the **-j** option.
- Message:** Error closing input file *filename*
Cause: The RCS file *filename* is compressed and an error occurred during decompression.
Action: The file is possibly corrupt. Try again with another copy of the file.
- Message:** Inconsistent date/time: *date*
Cause: You specified an argument with the **-d** option that was not a valid date and time.
Action: Make sure that the time you entered was correct, and try again.
- Message:** join pair incomplete
Cause: You specified the **-j** option, but the list of revisions provided as an argument was not in the correct format.
Action: Check the *DESCRIPTION* section for details on the format of the revision list used with the **-j** option.
- Message:** merge failed.: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** Missing date for **-d** option
Cause: You specified the **-d** option, but did not provide a date as its argument.
Action: Provide the missing date.

-
- Message:** No filename present for `-F` option.
Cause: You specified the `-F` option, but did not provide a file name as its argument.
Action: Provide the missing file name.
- Message:** No input file.
Cause: You failed to specify an input file on the command line.
Action: Provide the missing file name.
- Message:** Redefinition of `-option` option
Cause: You specified the `-d`, `-j`, `-s`, or `-w` option more than once on the command line.
Action: Specify each of the given options only once on a command line.
- Message:** Redefinition of revision number
Cause: You have specified more than one revision to be checked out. The `-f`, `-l`, `-p`, `-q`, `-r`, and `-u` options can each take a revision as an argument; however, only one option per command line may specify a revision.
Action: Specify, at most, a single revision.
- Message:** revision *rev* locked by *username*; use `co -r` or `rCS -u`
Cause: You specified a revision *rev* with the `-u` flag that is already locked by *username*.
Action: Either use the `-u` option of `rCS` to break the lock or the `-r` option of `co` to check out the given revision without attempting to unlock it.
- Message:** too many joins
Cause: The list of revisions that you provided as an argument to the `-j` option contained more than the maximum number of revisions that `co` can join.
Action: Try to join subsets of the list in consecutive calls to `co`.
- Message:** Unknown option "`-option`"
Cause: You specified an option that is not valid for `co`.
Action: Check the *DESCRIPTION* section for a list of valid `co` options.
- Message:** Write error when zeroing file *filename*: *system error*
Cause: See `syserror(3)`.
Action: See `syserror(3)`.
- Message:** writeable *filename* exists; checkout aborted.
Cause: You attempt to check out the file *filename*, but `co` found a writable version of the file, and refused to overwrite it.
Action: Make sure the specified file name was correct. Determine whether there have been any changes made to the writable version, delete or move it, and try again.

For a list of error messages common to all RCS utilities, see `rCSerror(3)`.

PORTABILITY

All UNIX systems.

The **-F**, **-G**, **-g**, **-I**, **-K**, **-O**, **-q**, and **-Y** options are extensions to traditional implementations of **co**.

NOTE

The **-d** option accepts no date before 1970.

Some unusual date formats may not be recognized.

Links to the RCS and working files are not preserved.

MPE/iX NOTES

The current MPE/iX implementation of **co** has the following limitations:

- Due to the fact that the comma (,) is not a valid character in MPE/iX file names, the traditional ,v naming convention is not currently implemented. You must have a subdirectory named RCS under your current directory or RCS will not work properly.
- For compatibility with traditional MPE/iX security features, you can only rename a file in an MPE/iX group if you own the file or if you have SM capability. Because this command attempts to rename files to which it may not have write access, this feature can interfere with the command's operation. For this reason, RCS archives should not be placed in an MPE/iX group, but rather should be located elsewhere in the hierarchical file system where the restriction does not apply. For example, the default *RCSPATH* places the archives in a subdirectory named RCS in the current working directory.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

ci(1), **ident(1)**, **racs(1)**, **racs-clean(1)**, **racsdiff(1)**, **racsmerge(1)**, **rlog(1)**,
racsfile(3)

NAME

: (colon) — do nothing, successfully

SYNOPSIS

: [*argument ...*]

DESCRIPTION

The **:** (colon) command simply yields an exit status of zero (success). This can be surprisingly useful, for example, when you are evaluating shell expressions for their side effects.

EXAMPLES

```
: ${VAR:="default value"}
```

sets *VAR* to a default value if and only if it is not already set.

DIAGNOSTICS

Since this command always succeeds, the only possible exit status is:

0 Successful completion.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh(1)** man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

NOTE

This is a special built-in command of the shell.

SEE ALSO

sh(1), **true(1)**

NAME

comm — compare sorted files and show differences

SYNOPSIS

comm [-123] *file1 file2*

DESCRIPTION

comm locates identical lines within files sorted in the same collating sequence, and produces three columns; the first contains lines found only in the first file, the second lines only in the second file, and the third lines which are in both files.

Options

comm accepts the following options:

- 1 suppresses lines that appear only in *file1*
- 2 suppresses lines that appear only in *file2*
- 3 suppresses lines that appear both in *file1* and *file2*

The options suppress individual columns; thus to list only the lines common to both files, use

```
comm -12
```

To find lines which are unique to one file or the other use

```
comm -3
```

Observe that

```
comm -123
```

displays nothing.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure because of an error opening or reading an input file or because of an invalid command line option.

Messages

Message: comm: file "*filename*": system error

Cause: See **syserror**(3).

Action: See **syserror**(3).

Message: strcoll error, cannot malloc space.
Cause: There are not enough free system resources to allocate string space.
Action: Free up more resources.

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **comm**.
Action: Check the *DESCRIPTION* section for a list of valid **comm** options.

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

cmp(1), **diff(1)**, **sort(1)**, **uniq(1)**

NAME

command — execute a simple command

SYNOPSIS

```
command [-p] command-name[argument...]
```

```
command [-v|-V] command-name
```

DESCRIPTION

command causes the shell to suppress its function lookup and execute the given *command-name* and *arguments* as though they made up a standard command line. In most cases, if *command-name* is not the name of a function, the results are the same as omitting **command**. If, however, *command-name* is a special built-in utility (see **sh**(1)), some unique properties of special built-ins do not apply:

- A syntax error in the utility does not cause the shell executing the utility to abort.
- Variable assignments specified with the special built-in utility do not remain in effect after the shell has executed the utility.

Options

command accepts the following options:

- p** searches for *command-name* using the default system *PATH*.
- V** writes a string indicating how the shell interprets *command-name*. If *command-name* is a utility, regular built-in utility, or an implementation-provided function found using the *PATH* variable, the string identifies it as such and includes the absolute path name. If *command-name* is an alias, function, special built-in utility, or reserved word, the string identifies it as such and includes its definition if it is an alias.
- v** writes a string indicating the path name or command that the shell uses to invoke *command-name*.

EXAMPLES

Typically, you use **command** when you have a command that may have the same name as a function. For example, here's a definition of a **cd** function that not only switches to a new directory, but also uses **lc** to list the contents of that directory.

```
function cd {
    command cd $1
    lc
}
```

Inside the function, we use **command** to get at the real **cd**. If we didn't do this, the **cd** function would call itself in an infinite recursion.

ENVIRONMENT VARIABLES

command uses the following environment variable:

PATH

contains a list of directories for **command** to use when searching for *command-name* except as described under the **-p** option.

DIAGNOSTICS

If you specified **-v**, possible exit status values are:

- 0 Successful completion.
- 1 **command** was unable to find *command-name* or an error occurred.
- 2 Failure due to invalid command line argument.

If you did not specify **-v**, possible exit status values are:

- 126 **command** found *command-name*, but failed to invoke it.
- 127 An error occurred in the **command** utility or it was unable to find *command-name*.

Otherwise, the exit status of **command** is the exit status of *command-name*.

Messages

- Message:** *command-name*: not found
Cause: You specified a *command-name* that **command** was unable to find.
Action: Check that the *command-name* exists, was spelled properly, and that you have the appropriate permissions.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **command**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **command** options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

sh(1)

NAME

compress — Lempel-Ziv compression of a file

SYNOPSIS

compress [-cDdfVv] [-b *bits*] [*file* ...]

DESCRIPTION

compress compresses each input *file* using Lempel-Ziv compression techniques. If you do not specify any input files, **compress** reads data from the standard input and writes the compressed result to the standard output.

On UNIX and POSIX-compliant systems, the output files have the same names as the input files but with a .Z suffix. For example, abc is compressed into abc.Z. If the .Z file already exists and you did not specify the **-f** option, **compress** gives an error and asks you whether or not it should overwrite the existing file.

compress uses the modified Lempel-Ziv algorithm (described in *A Technique for High Performance Data Compression*, Terry A. Welch, *IEEE Computer*, vol. 17, no. 6 (June 1984), pp. 8-19). **compress** first replaces common substrings in the file by 9-bit codes starting at 257. After it reaches code 512, **compress** begins with 10-bit codes, and continues to use more bits until it reaches the limit set by the **-b** option.

After attaining the *bits* limit, **compress** periodically checks the compression ratio. If it is increasing, **compress** continues to use the existing code dictionary. However, if the compression ratio decreases, **compress** discards the table of substrings and rebuilds it from scratch. This allows the algorithm to compensate for files, such as archives, where individual components have different information content profiles.

Options

compress accepts the following options:

- b** *bits* limits the maximum number of bits of compression to *bits*. The value *bits* may be an integer from 9 to 16. The default is 16.
- c** writes the output to the standard output. When you use this option, you can only specify one *file* on the command line.
- D** performs an extra degree of compression on files such as sorted dictionaries where consecutive lines normally have many characters in common.
- d** decompresses argument files instead of compressing them. This works by overlaying the **compress** program with the **uncompress** program. For this to work, **uncompress** must be available somewhere in your search path (given by the *PATH* environment variable). Decompressing files this way is slower than calling **uncompress** directly.

-
- f** forces compression even if the resulting file is larger or the output file already exists. When you do not specify this option, files which are larger after compression are not compressed. **compress** does not print an error message if this happens.
 - v** prints the version number of **compress**.
 - v** prints statistics giving the amount of compression achieved. Statistics give the name of each file compressed and the compression ratio, expressed as a percentage. If the file resulting from compression is larger than the original, the compression ratio is negative.

ENVIRONMENT VARIABLES

compress uses the following environment variable:

PATH contains a list of directories for **compress** to search when looking for the **uncompress** utility.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.
- 2 One or more files were not compressed because the compressed version was larger than the original.

Messages

- Message:** (-D) same count exceeded - aborting
Cause: When using the **-D** option, **compress** encountered a line where more than the first 255 characters were identical to those of the previous line.
Action: Make sure that no two adjacent lines in a file being with **-D** have more than the first 255 characters in common.
- Message:** Bits must be between *num1* and *num2*
Cause: You specified a value for the maximum number of bits of compression with the **-b** option that fell outside of the range *num1* to *num2*.
Action: Specify a value that falls in the *num1* to *num2* range.
- Message:** can't stat file "*filename*": system error
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

Message: exec "uncompress" : *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: file "*filename*" : *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: *filename* has *num* other links: unchanged
Cause: You specified a file that had more than one link. **compress** will not compress such a file.
Action: Remove the additional links, or use **-f** option.

Message: *filename* not a regular file: unchanged
Cause: You specified a file that was not a regular file. **compress** will not compress directories, FIFOs, or other such files.
Action: Make sure that the specified *filename* is a regular file.

Message: input file "*filename*" : *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: insufficient memory
Cause: There were not enough free system resources to perform the specified operation.
Action: Free up more resources.

Message: no space for compression tables There were not enough free system resources to allocate to compression tables.
Action: Free up more resources. "Option *-option* argument missing"
Cause: You did not provide an argument for *-option*.
Action: Provide the missing argument.

Message: output file "*filename*" : *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: output path or file name too long
Cause: When the .Z extension was appended to the original file's name, it resulted in an output path or file name that was too long for the file system to handle.
Action: Rename the original file with a shorter name.

Message: tempfile: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

- Message:** unable to create tempfile name
Cause: **compress** was unable to create a temporary file in the directory named by *TMPDIR*, the /tmp directory, or the current directory.
Action: Make sure that you have appropriate permissions to create a temporary file in one of these three directories.
- Message:** unknown error
Cause: An unknown compression error occurred.
Action: Contact your system manager.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **compress**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **compress** options.

LIMITS

This implementation of **compress** is limited to a maximum of 16 bit compression.

PORTABILITY

A binary-compatible version of **compress**, with more options including the ability to compress in place, is often found on UNIX systems.

The **-D** option is an extension to traditional implementations of **compress**.

MPE/iX NOTES

The current MPE/iX implementation of **compress** converts non-byte files to byte stream files before compressing them. File characteristics like file code, record size, and so forth are not preserved by this conversion. When you decompress the compressed file, it is written as a byte stream file.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

cpio(1), **pack(1)**, **pax(1)**, **tar(1)**, **uncompress(1)**, **unpack(1)** **zcat(1)**

NAME

continue — skip to next iteration of loop in shell script

SYNOPSIS

continue [*n*]

DESCRIPTION

continue skips to the next iteration of an enclosing **for**, **select**, **until**, or **while** loop in a shell script. If a number *n* is given, execution continues at the loop-control of the *n*th enclosing loop. The default value of *n* is 1.

DIAGNOSTICS

The exit status of **continue** is always zero.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh**(1) man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

NOTE

This is a special built-in command of the shell.

SEE ALSO

break(1), **sh**(1)

NAME

cp — copy files

SYNOPSIS

```
cp [-c|f|imp] file1 file2
cp [-f|imp] file ... directory
cp -R [-f|imp] source... directory
cp -r [-f|imp] source... directory
```

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

cp copies files to a target named by the last argument on its command line. If the target is an existing file, **cp** overwrites it; if it does not exist, **cp** creates it. If the target file already exists and does not have write permission, **cp** denies access and continues with the next copy.

If you specify more than two path names, the last path name (that is, the target) must be a directory. If the target is a directory, **cp** copies the sources into that directory with names given by the final component of the source path name.

Options

cp accepts the following options:

- f** attempts to replace files which do not have write permission.
- i** asks you if you want to overwrite an existing file, whether or not the file is read-only.
- m** sets the modify and access time of each destination file to that of the corresponding source file. Normally, **cp** sets the modification time of the destination file to the present.
- p** preserves the modify and access times (like the **-m** option); in addition, it preserves file mode, owner, and group owner, if possible.
- R** *clones* the source trees. **cp** copies all the files and subdirectories specified by *source...* into *directory*, making careful arrangements to duplicate special files (FIFO, block special, character special).
- r** *clones* the source trees, but makes no allowances for special files (FIFO, block special, character special). Consequently, **cp** attempts to read from a device rather than duplicate the special file. This is similar to, but less useful than, the preferred **-R**.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - an argument had a trailing slash (/) but was not a directory
 - unable to find a file
 - unable to open an input file for reading
 - unable to create or open an output file
 - a read error occurred on an input file
 - a write error occurred on an output file
 - the input and output files were the same file
 - encountered a fatal error when using **-r** or **-R**.
Possible fatal **-r** or **-R** errors include:
 - inability to access a file
 - inability to change permissions on a target file
 - inability to read a directory
 - inability to create a directory
 - a target which is not a directory
 - source and destination directories are the same.
- 2 Failure due to any of the following:
 - an invalid command line option
 - too few arguments on the command line
 - a target that should be a directory but isn't
 - no space left on target device
 - out of memory to hold the data to be copied
 - inability to create a directory to hold a target file

Messages

Message: cannot allocate I/O buffer: *system error*

Cause: See **syserror(3)**.

Action: See **syserror(3)**.

Message: cannot allocate target string

Cause: **cp** has no space to hold the name of the target file.

Action: Free up more system resources.

Message: cannot create parent directory for target "*name*"

Cause: An error occurred while trying to create the parent directory of the specified target file.

Action: Make sure that you have permissions to create the directory.

-
- Message:** cannot find file "*filename*"
Cause: You specified a *filename* that does not exist.
Action: Check the path and spelling of *filename*.
- Message:** cannot mkdir "*pathname*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** cannot open file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Cannot reset permissions on file "*filename*": *system error*"
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Cannot reset times on file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Cannot reset uid or gid on file "*filename*": *system error*"
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** copy of *filename* failed (CIERR *num*)
Cause: An error occurred while using the MPE/iX CI **COPY** command to copy a non-byte stream file.
Action: Refer to the *MPE/iX Reference Supplement (32650-90353)* for more information.
- Message:** fifo "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** no space on device for file "*filename*"
Cause: You attempted to copy a file to *filename* on a device that has no space for it.
Action: Free up space on the target device or copy the file to another device.
- Message:** read error on file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** recursive copy to directory "*pathname*"
Cause: You tried to recursively copy a directory to itself.
Action: Choose a different *pathname*.

-
- Message:** source *"name"* and target *"name"* are identical
Cause: You specified source and target files that are actually the same file (for example, because of links).
Action: No further action is required.
- Message:** special file *"filename"* system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** stat error for *"filename"* : system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** target file *"filename"* : system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** target *"pathname"* is not a directory
Cause: When recursively copying multiple files using the **-r** or **-R** option, the target must be a directory. You specified a target *pathname* that is not a directory.
Action: Check spelling of target *pathname*.
- Message:** target *"pathname"* must be a directory
Cause: You attempted to copy two or more files but the target indicated by *pathname* was not a directory.
Action: When copying (or moving) two or more files, ensure that the final *name* on the command line is a directory.
- Message:** target *"pathname"* must exist
Cause: The destination directory must exist for this utility to work.
Action: Check the command line arguments. You may need to create the target directory.
- Message:** Unknown option *"-option"*
Cause: You specified an option that is not valid for **cp**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **cp** options.
- Message:** unreadable directory *"pathname"* : system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** write error on file *"filename"* : system error
Cause: See **syserror**(3).
Action: See **syserror**(3).

PORTABILITY

POSIX.2. *x/OPEN* Portability Guide 4.0. All UNIX systems.

The **-f** and **-m** options are extensions to the POSIX standard.

MPE/iX NOTES

When copying byte stream files, **cp** performs in the POSIX-compliant manner described in this man page. When copying non-byte stream files, **cp** calls the MPE/iX CI **COPY** command to perform the task. See the *MPE/iX Reference Supplement* (32650-90353) for details on how the **COPY** command works.

On MPE/iX, **cp** is available as both a built-in shell utility and an external utility.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

cat(1), **cpio(1)**, **mv(1)**, **rm(1)**

NAME

cpio — archiver to copy and back up files

SYNOPSIS

```
cpio -o [-aBcvyz] [-C blocksize] [-O file] [-V volpat]  
cpio -i [-BbcdfmrsStuv6qyz] [-C blocksize] [-I file] [-V volpat] [pattern ...]  
cpio -p [-aBdlmruv] directory
```

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

The **cpio** command manipulates files called *cpio archives*. A **cpio** archive is a concatenation of files and directories preceded by a header giving the file name and other file system information. With **cpio**, you can create a new archive, extract contents of an existing archive, list archive contents, and copy files from one directory to another.

Options

Every call to **cpio** must specify one and only one of the following *selector* options:

-i reads an existing archive (created with the **-o** option) from the standard input. Unless you specify the **-t** option, **cpio** extracts all files matching one or more of the given *pattern* arguments from the archive. Patterns are the same as those used by file name generation (see **sh**(1)). When you do not specify a *pattern* argument, the default pattern ***** is used; as a result, **cpio** extracts all files.

-o writes a new archive to the standard output, using the list of files read from the standard input. Such a list might be produced by the **ls** or **find** commands. For example,

```
ls . | cpio -o >arch
```

uses **ls** to list the files of the current directory, then pipes this list as input to **cpio**. The resulting archive contains the contents of all the files, and is written to **arch**.

-p is shorthand for

```
cpio -o | (cd directory; cpio -i)
```

where the **cpio -i** is performed in the given *directory*. You can use this option to copy entire file trees.

Consult the synopsis lines to determine which of the following additional options can be applied with a particular selector option.

-
- a** resets the access time of each file accessed for copying to the archive to what it was before the copy took place.
 - B** uses buffers of 5120 bytes for input and output rather than the default 512 byte buffers.
 - b** causes 16-bit words to be swapped within each longword and bytes to be swapped within each 16-bit word of each extracted file. This facilitates the transfer of information between different CPU architectures. This is equivalent to specifying both the **-s** and **-S** options.
 - C** *blocksize*
sets the buffer size to a specified blocksize, rather than the default 512 byte buffers.
 - c** reads and writes header information in ASCII form. Normally, **cpio** writes the header information in a compact binary format. This option produces an archive more amenable to transfer through non-binary streams (such as some data communications links) and is highly recommended for those moving data between different processors, such as between a UNIX system and a DOS PC.
 - d** forces the creation of necessary intermediate directories when they do not already exist.
 - f** inverts the sense of pattern matching. More precisely, **cpio** extracts a file from the archive if and only if it does *not* match any of the *pattern* arguments.
 - I** *file* causes input to be read from the specified file, rather than stdin.
 - l** gives permission to create a link to a file rather than making a separate copy.
 - m** resets the modification time of an output file to the modification time of the source file. Normally, when **cpio** copies data into a file, it sets the modification time of the file to the time at which the file is written.
 - O** *file* causes output to be written to the specified file, rather than stdout.
 - q** assumes all created files are ASCII text. On UNIX and POSIX-compliant systems, this means that any `\r` (carriage return) characters are stripped, and only the `\n` (new-lines) are retained.

It might be desirable to have this option work on output also, converting text to a system-independent format; however, due to the format of an archive file, this would (unacceptably) require all files to be read twice.
 - r** provides an interactive mechanism for selecting and renaming particular files. For each file processed, **cpio** displays the name before copying it to its new location. At this point, you may type in a new name for the file. If you enter an empty line, the file is skipped.

-
- S** for portability reasons, swaps pairs of 16-bit words within longwords only when extracting files. This option does not affect the headers.
 - s** for portability reasons, swaps pairs of bytes within each 16-bit word only when extracting files. **-s** does not affect the headers.
 - t** prevents files extraction, producing instead a table of file names contained in the archive. See the description of the **-v** option.
 - u** copies an archive file to a target file even if the target is newer than the archive. Normally, **cpio** does not copy the file.
 - V** *volpat*
provides automatic multi-volume support. **cpio** writes output to files, the names of which are formatted using *volpat*. The current volume number replaces any occurrence of # in *volpat*. When you invoke **cpio** with this option, it asks for the first number in the archive set, and waits for you to type the number and a carriage return before it proceeds with the operation. **cpio** issues the same sort of message when a write error or read error occurs on the archive; the reasoning is that this kind of error means that **cpio** has reached the end of the volume and should go on to a new one.
 - v** provides more verbose information than usual. **cpio** prints the names of files as it extracts them from or adds them to archives. When you specify both **-v** and **-t**, **cpio** prints a table of files in a format similar to that produced by the **ls -l** command.
 - y** when used with **-v**, does not ask for a volume number to begin with, but does ask if it gets a read or write error.
 - z** performs Lempel-Ziv compression. Output is always a 16-bit compression. On input, any compression up to 16-bit is acceptable. This option should be equivalent to the following pipelines:

```
cpio -ocvz <---> cpio -ocv | compress -b 16
cpio -icvz <---> uncompress | cpio -icv
```
 - 6** is supposed to understand 6th edition UNIX **cpio** archives, but is not currently implemented nor deemed necessary.

The byte and word swapping done by the **-b**, **-s**, and **-S** options is effective only for the file data written. With or without the **-c** option, header information is always written in a machine-invariant format.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** -6 not supported
Cause: You specified the **-6** option, which is not currently implemented.
Action: Do not use the **-6** option.
- Message:** -I: Must specify -i option
Cause: You specified the **-I** option, but did not specify the **-i** option.
Action: To use the **-I** option, you must specify the **-i** option.
- Message:** -O: Must specify -o option
Cause: You specified the **-O** option, but did not specify the **-o** option.
Action: To use the **-O** option, you must specify the **-o** option.
- Message:** -r option disabled with -p
Cause: You specified the interactive renaming option (**-r**) with the pass option (**-p**). **cpio** does not allow this combination.
Action: Do not specify the **-r** option on the same command line with the **-p** option.
- Message:** Directory "*pathname*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** I/O buffer allocation: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Insufficient memory: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Must specify one of -i, -o, or -p
Cause: You must specify one of the **-i**, **-o**, or **-p** options. You failed to do so.
Action: Specify one of the required options.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **cpio**.
Action: Check the *DESCRIPTION* section for a list of valid **cpio** options.

cpio may also produce several of the error messages listed on the **pax**(1) man page. See that man page for more details.

PORTABILITY

*x*OPEN Portability Guide 4.0. All non-Berkeley UNIX systems after Version 7.

The **-q**, **-v**, **-y**, and **-z** options are extensions to traditional implementations of **cpio**.

MPE/iX NOTES

The current MPE/iX implementation of **cpio** has the following limitations:

- It converts non-byte stream files to byte stream files before archiving them. File characteristics like file code, record size, and so forth are not preserved by this conversion. When you extract files from an archive, they are written as byte stream files.
- It displays file owner names as 17 character fields in the form *username.accountname* and group names as 8 character fields in the form *groupname*.
- Because this release of MPE/iX does not provide the `lstat()` API, **cpio** cannot return information on the link itself. It attempts to determine when a symbolic link has been referenced, but can only return the information on the target of the link, rather than the link itself.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

compress(1), **cp**(1), **dd**(1), **find**(1), **ls**(1), **mv**(1), **pax**(1), **tar**(1), **uncompress**(1), **cpio**(2), **tar**(2)

NAME

csplit — split a text file, according to criteria

SYNOPSIS

csplit [-Aaks] [-f *prefix*] [-n *number*] *file* *arg* *arg* ...

DESCRIPTION

csplit takes a text *file* as input and breaks up its contents into pieces, based on criteria given by the *arg* value(s) on the command line. For example, you can use **csplit** to break up a text file into chunks of ten lines each, then save each of those chunks in a separate file. See the subsection *Splitting Criteria* for more details. If you specify - as the *file* argument, **csplit** uses the standard input.

The files created by **csplit** normally have names of the form

xxnumber

where *number* is a two digit decimal number which begins at zero and increments by one for each new file that **csplit** creates.

csplit also displays the size, in bytes, of each file that it creates.

Options

csplit accepts the following options:

- A uses uppercase letters in place of numbers in the number portion of created file names. This generates names of the form xxAA, xxAB, and so on.
- a uses lowercase letters in place of numbers in the number portion of created file names. This generates names of the form xxaa, xxab, and so on.
- f *prefix* specifies a *prefix* to use in place of the default xx when naming files. If *prefix* causes a file name longer than NAME_MAX bytes, an error occurs and **csplit** exits without creating any files.
- k leaves all created files intact. Normally, when an error occurs, **csplit** removes files that it has created.
- n *number* specifies the number of digits in the number portion of created file names.
- s suppresses the display of file sizes.

Splitting Criteria

csplit processes the *args* on the command line sequentially. The first argument *breaks off* the first chunk of the file, the second argument *breaks off* the next chunk (beginning at the first line remaining in the file), and so on. Thus each chunk of the file begins with the first line remaining in the file and goes to the line given by the next *arg*.

arg values may take any of the following forms:

/regexp/ takes the chunk as all the lines from the current line up to *but not including* the next line that contains a string matching the regular expression *regexp*. *regexp* is a basic regular expression (see **regexp(3)**). After **csplit** has obtained the chunk and written it to an output file, it sets the current line to the line that matched *regexp*.

/regexp/offset is the same as the previous criterion, except that the chunk goes up to but not including the line that is a given *offset* from the first line containing a string that matches *regexp*. The *offset* may be a positive or negative integer. After **csplit** has obtained the chunk and written it to an output file, it sets the current line to the line that matched *regexp*.

Note: This current line is the first one that was not part of the chunk just written out.

%regexp% is the same as */regexp/*, except that **csplit** does not write the chunk to an output file. It simply skips over the chunk.

%regexp%offset is the same as */regexp/offset*, except **csplit** does not write the chunk to an output file.

linenumber obtains a chunk beginning at the current line and going up to but not including the *linenumber*th line. After **split** writes the chunk to an output file, it sets the current line to *linenumber*.

{number} repeats the previous criterion *number* times. If it follows a regular expression criterion, it repeats the regular expression process *number* more times. If it follows a *linenumber* criterion, **csplit** splits the file every *linenumber* lines, *number* times, beginning at the current line. For example,

```
csplit file 10 {10}
```

obtains a chunk from line 1 to line 9, then every 10 lines after that, up to line 109.

Errors occur if any criterion tries to *grab* lines beyond the end of the file, if a regular expression does not match any line between the current line and the end of the file, or if an *offset* refers to a position before the current line or past the end of the file.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - because **csplit** was unable to open the input or output files
 - a write error on the output file.
- 2 Failure due to any of the following:
 - unknown command line option
 - the *prefix* name was missing after **-f**
 - the *number* of digits was missing after **-n**
 - the input *file* was not specified
 - no *arg* values were specified
 - the command ran out of memory
 - an *arg* was invalid
 - the command found end-of-file prematurely
 - a regular expression in an *arg* was badly formed
 - a line offset/number in an *arg* was badly formed
 - a {*number*} repetition count was misplaced or badly formed
 - too many file names were generated when using **-n**
 - generated file names would be too long.

Messages

- Message:** Badly formed line number "*linenumber*"
Cause: You specified a *linenumber* that was not a valid integer.
Action: Specify a valid integer for *linenumber*.
- Message:** Badly formed line offset in "*offset*"
Cause: You specified an *offset* that was not a valid integer.
Action: Specify a valid integer for *offset*.
- Message:** Badly formed regular expression *regexp*
Cause: You specified the regular expression *regexp*, but it did not contain a closing / or %.
Action: Provide the missing / or %.
- Message:** Badly formed {*repeat*} count "*number*"
Cause: You specified a {*number*} that was not a valid integer.
Action: Specify a valid integer for *number*.

-
- Message:** cannot create temporary file: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** csplit argument must be one of ...
Cause: You specified an argument to **csplit** that did not look like a regular expression, line number, or repeat count.
Action: Check the syntax of your command line, correct any mistakes, and re-enter it.
- Message:** error in regular expression *regexp: regular expression error*
Cause: See **regerror**(3).
Action: See **regerror**(3).
- Message:** Generated filenames would be too long
Cause: You specified an argument for the **-n** option that when combined with the length of the specified *prefix* resulted in a generated file name that was longer than the file system permits.
Action: Specify a shorter *prefix* or a lower value for the argument to the **-n** option.
- Message:** input file "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** insufficient memory: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** Misplaced {*number*} repetition count
Cause: You specified a {*number*} repetition count before specifying the splitting criteria, or you specified the count without specifying any criteria at all.
Action: Make sure that any {*number*} arguments follow the appropriate criteria specification on the command line.
- Message:** missing input file
Cause: You did not specify an input file.
Action: Specify an input file.
- Message:** Missing number of digits
Cause: You specified the **-n** option but did not provide an argument to indicate the number of digits to use when generating file names.
Action: Provide the missing argument.
- Message:** Missing prefix file name
Cause: You specified the **-f** option but did not provide a prefix as an argument.
Action: Provide the missing prefix.

-
- Message:** need at least one section argument
Cause: You did not specify any splitting criteria.
Action: Specify at least one argument which defines splitting criteria.
- Message:** *num1* digits specified (*limit* allowed)
Cause: You specified an argument to the **-n** option that was greater than the number of digits allowed by **csplit** (that is, *limit*).
Action: Specify a smaller *num1* or a larger *limit*.
- Message:** output file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** too many file names generated, use "-n" option
Cause: You specified a set of criteria that created more files than **csplit** was able to generate names for, given the number of digits in the numeric portion of file names.
Action: Specify criteria that creates fewer files, or use the **-n** option to increase the number of digits in the numeric portion of file names.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **split**.
Action: Check the *DESCRIPTION* section for a list of valid **csplit** options.
- Message:** Warning: premature EOF at "*arg*"
Cause: **csplit** reached the end-of-file before completing the search for *arg*.
Action: Specify criteria that will not cause **csplit** to try searching past the end of the file.
- Message:** write error on file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

The **-A** and **-a** options are extensions to the POSIX standard.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

awk(1), **sed(1)**, **regexp(3)**

NAME

ctags — produce tags file for **ex**, **more**, and **vi**

SYNOPSIS

ctags [**-aBFwx**] [**-f tagfile**] *sourcefile* ...

DESCRIPTION

By default, **ctags** generates a file named `tags` in the current directory which summarizes the C function, macro, and `typedef` definitions found in the *sourcefiles* named on the command line. See **tags(2)** for a description of the format of the `tags` file. You can access this file with the **-t name** option in **ex**, **more**, and **vi**, the command

```
:tag name
```

in **ex** and **vi**, and the command

```
:tname
```

in **more**. The idea is that you tell the utility which function you want to look at, and it checks the `tags` file to determine which source file contains the function.

ctags makes special provision for the function `main()` which may occur in several source files. The `tags` file contains an entry for the first `main()` routine found. For all occurrences of `main()` (including the first), the `tags` file also contains an entry for *Mname* where *name* is the name of the input *sourcefile*, with the `.c` suffix and any leading path name components removed. For example, a `tags` file created for a C source code file named `foo.c` would contain an entry for `Mfoo` which represents the `main()` routine in `foo.c`.

ctags uses **sort(1)** to sort the file by tag name, according to the POSIX locale's collation sequence.

Options

ctags accepts the following options:

- a** appends output to the existing `tags` file rather than overwriting the file.
- B** produces a `tags` file that searches backward from the current position to find the pattern matching the tag.
- F** searches for tag patterns in the forward direction (default).
- f tagfile**
generates a file named *tagfile* rather than the default `tags`.
- w** suppresses warning messages.

- x** produces a human-readable report on the standard output. The report gives the definition name, the line number of where it appears in the file, the name of the file in which it appears, and the text of that line. **ctags** arranges this output in columns and sorts it in order by tag name according to the current locale's collation sequence. This option does not produce a tags file.

FILES

ctags generates the following file:

tags output tags file

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** cannot allocate buffer: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** cannot create temp file: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** cannot determine NAME_MAX: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** cannot open temp file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Don't know how to process "*filename*"
Cause: You specified an input file that did not have a .c or .h extension.
Action: Only specify input files with .c or .h extensions.
- Message:** file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Option *-option* argument missing
Cause: You did not provide an argument for *-option*.
Action: Provide the missing argument.

Message: pipe to command "*cmd*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: tag file "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: tmp file "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: unable to pipe to sort command: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **ctags**.
Action: Check the *DESCRIPTION* section for a list of valid **ctags** options.

Message: Warning: *ident* already defined in *filename*.
Cause: You defined the identifier *ident* multiple times. The occurrence which produced this warning is ignored.
Action: Remove the extra definitions of *ident*.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. 4.2 BSD UNIX and up.

The only language understood by this version of **ctags** is C.

The **-B**, **-F**, and **-w** options are MPE/iX extensions to the POSIX standard.

NOTE

Recognizing a function definition in C source code can be somewhat difficult. Since **ctags** does not know which C preprocessor symbols are defined, there may be some misplaced function definition information if sections of code within `#if...#endif` are not complete blocks.

ctags invokes the **sort** command internally.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

ctags(1)

MPE/iX Shell and Utilities

ctags(1)

SEE ALSO

more(1), sort(1), vi(1), tags(2)

NAME

cut — selectively display fields or characters from input lines

SYNOPSIS

```
cut -b list [-n] [file...]  
cut -c list [file...]  
cut -f list [-d char] [-s] [file...]
```

DESCRIPTION

cut reads input from *files* and selectively copies sections of the input lines to the standard output. If you do not specify any *files*, or you specify a file named **-**, **cut** reads from the standard input.

Options

cut accepts the following options:

- b list** invokes byte position mode. After this comes a list of the byte positions you want to display. This list may contain multiple byte positions, separated by commas (,) or blanks or ranges of positions separated by dashes (-); since the list must be one argument, shell quoting is necessary if blanks are used. You may combine these to allow selection of any byte positions of the input.
- c list** invokes character position mode. After this comes a list of character positions to retain in the output. This list may contain multiple character positions, separated by commas (,) or blanks or ranges of positions separated by a dash (-); since the list must be one argument, shell quoting is necessary if blanks are used. You may combine these to allow selection of any character positions of the input.
- d char**
specifies *char* as the character that separates fields in the input data; by default, this is the horizontal tab.
- f list** invokes field delimiter mode. After this comes a list of the fields you want to display. You specify ranges of fields and multiple field numbers in the same way you specify ranges of character positions and multiple character positions in **-c** mode.
- n** does not split characters. If the low byte in a selected range is not the first byte of a character, **cut** extends the range downward to include the entire character; if the high byte in a selected range is not the last byte of a character, **cut** limits the range to include only the last entire character before the high byte selected. If **-n** is selected, **cut** does not list ranges that do not encompass an entire character and these ranges do not cause an error.

- s** does not display lines that do not contain a field separator character. Normally, **cut** displays lines that do not contain a field separator character in their entirety.

EXAMPLES

```
cut -f 2,2 -d " " /etc/profile
```

displays the second space-delineated field in the system profile.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** bad list for **-f**, **-b**, or **-c** option [*list*]
Cause: You specified a list for the **-f**, **-b**, or **-c** option that contained non-numeric entries.
Action: Specify a list that contains only numeric entries.
- Message:** badly formed range in list [*list*]
Cause: You specified a list that contained a range that was not in the form:
num1-num2
Action: Re-enter the command line using the proper syntax for a range.
- Message:** Bad range "*num1-num2*" in list
Cause: You specified a list containing the range *num1-num2* where *num2* was less than *num1*. Ranges must be specified with the lower value first.
Action: Re-enter the command line, making sure to list the lower value first when specifying the range.
- Message:** Field delimiter specified by **-d** must be one character
Cause: You specified a field delimiter (as an argument to the **-d** option) that was more than one character long.
Action: Specify a single character field delimiter.
- Message:** file "*filename*": system error
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** input file "*filename*": system error
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

-
- Message:** Missing character after `-d`
Cause: You specified the `-d` option, but did not provide a field separator character as its argument.
Action: Provide the missing field separator character.
- Message:** Must specify `"-f"`, `"-b"` or `"-c"` option
Cause: You did not specify any of the `-f`, `-b`, or `-c` options.
Action: Specify one of the three options.
- Message:** no fields specified in list `[list]`
Cause: `cut` did not recognize anything in `list` as indicating a field.
Action: Check the syntax of the list and re-enter the command.
- Message:** Option `-option` argument missing
Cause: You did not provide an argument for `-option`.
Action: Provide the missing argument.
- Message:** Out of memory
Cause: `cut` was unable to allocate the required system resources for internal buffers.
Action: Free up more system resources and try again.
- Message:** Unknown option `"-option"`
Cause: You specified an option that is not valid for `cut`.
Action: Check the *DESCRIPTION* section of this man page for a list of valid `cut` options.
- Message:** write error on standard output: *system error*
Cause: See `syserror(3)`.
Action: See `syserror(3)`.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. UNIX System V.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

`paste(1)`, `uname(1)`

NAME

date — set and display date and time

SYNOPSIS

date [-cu] [*timespec*]

date [-cu] [+*format*]

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

date either displays the operating system's idea of the current date and time, or sets it to a new value. The following example shows the default format of the date:

```
Wed Feb 26 14:01:43 EST 1986
```

Options

date accepts the following options:

-c sets or displays the date and time according to Greenwich Mean Time (Coordinated Universal Time) using CUT as the time zone name.

-u sets or displays the date and time according to Greenwich Mean Time (Coordinated Universal Time) using GMT as the time zone name.

Setting Date and Time

date also accepts an argument in one of two forms. If the argument does not begin with +, **date** assumes it is a *timespec* of the form

```
[ [ [ cc ]yy ]mm ]dd ]hhmm [.ss ]
```

where *cc* is the optional first 2 digits of the year, *yy* is the optional last 2 digits of the year, *mm* is the optional number of the month (01–12), *dd* is the optional day of the month, *hh* is the hour in 24 hour format (required), *mm* is the minutes (required), and *ss* is the optional seconds. **date** uses these values to set the date and time.

Note: You must specify the hours and the minutes; other arguments are optional.

Displaying Date and Time

If the argument to **date** begins with a + character, **date** uses *format* to display the date. **date** writes all characters in *format*, with the exception of the % and the character which immediately follows it, directly to the standard output. After **date** exhausts the *format* string, it outputs a newline character. The % character introduces a special format field similar to the printf() function in the C library (see *Field Descriptors*).

Field Descriptors

date recognizes the following field descriptors:

- %A the full weekday name in the current locale (for example, Sunday, in English).
- %a the abbreviation for the weekday in the current locale (for example, Sun, in English).
- %B the full month name in the current locale (for example, February, in English).
- %b the abbreviation for the month name in the current locale (for example, Feb, in English).
- %C the first two digits of the year (00 to 99).
- %c the appropriate representation of the date and time in the current locale.
- %D the date in the form *mm/dd/yy*.
- %d the two-digit day of the month as a number (01 to 31).
- %e the day of the month in a two-digit, right-justified, blank-filled field (1 to 31).
- %H the hour in the 24-hour clock representation (00 to 23).
- %h the same as %b.
- %I the hour in the 12-hour clock representation (01 to 12).
- %j the numeric day of the year (001 to 366).
- %M the minute (00 to 59).
- %m the month number (01 to 12).
- %n a newline character.
- %p the equivalent of AM or PM in the current locale.
- %r the 12-hour time in the current locale's equivalent of AM/PM notation (11:53:29 AM in the POSIX locale).
- %S the seconds (00 to 61). Note that there is an allowance for two leap seconds.
- %T the 24-hour time (14:53:29).
- %t a tab character.
- %U the week number in the year, with Sunday being the first day of the week (00 to 53). All days before the first Sunday of the new year are in week 0.

%u	the weekday number with Monday being 1 and Sunday being 7.
%V	the week number in the year, with Monday being the first day of the week (01 to 53). If the week containing January 1 has four or more days in the new year, it is week 1 of the new year; otherwise it is week 53 of the previous year.
%W	the week number in the year, with Monday being the first day of the week (00 to 53). All days before the first Monday of the new year are in week 0.
%w	the weekday number, with Sunday being 0 and Saturday being 6.
%X	the appropriate time representation in the current locale.
%x	the appropriate date representation in the current locale.
%Y	the year.
%y	the two-digit year (offset from %C).
%Z	the time zone name (for example, EDT).
%%	a percent-sign character.

The **date** command also supports the following modified field descriptors to indicate a different format as specified by the locale indicated by *LC_TIME*. If the current locale does not support a modified descriptor, **date** uses the unmodified field descriptor value.

%EC	the name of the base year (period) in the current locale's alternate representation.
%Ec	the current locale's alternate date and time representation.
%Ex	the current locale's alternate date representation.
%EY	the full alternate year representation.
%Ey	the offset from %EC (year only) in the current locale's alternate representation.
%Od	the day of month using the current locale's alternate numeric symbols .
%Oe	the day of month using the current locale's alternate numeric symbols in a two-character, right-justified, blank-filled field.
%OH	the hour (24-hour clock) using the current locale's alternate numeric symbols.
%OI	the hour (12-hour clock) using the current locale's alternate numeric symbols.
%OM	the minutes using the current locale's alternate numeric symbols.
%Om	the month using the current locale's alternate numeric symbols.
%OS	the seconds using the current locale's alternate numeric symbols.

-
- %OU the week number of the year (with Sunday as the first day of the week) using the current locale's alternate numeric symbols.
 - %Ou the weekday number using the current locale's alternate numeric symbols with Monday being 1 and Sunday being 7.
 - %OV the week number in the year using the current locale's alternate numeric symbols, with Monday being the first day of the week. If the week containing January 1 has four or more days in the new year, it is week 1 of the new year; otherwise it is week 53 of the previous year.
 - %OW the week number of the year (with Monday as the first day of the week) using the current locale's alternate numeric symbols.
 - %Ow the weekday as a number using the current locale's alternate numeric symbols with Sunday being 0 and Saturday being 6.
 - %Oy the year (offset from %C) using the current locale's alternate numeric symbols.

EXAMPLES

The command

```
date '+%a %b %e %T %Z %Y'
```

produces the date in the default format. For example,

```
Wed Feb 26 14:01:43 EST 1986
```

ENVIRONMENT VARIABLES

date uses the following environment variable:

- TZ* gives the time zone for **date** to use when displaying the times. This is ignored if you specify either the **-c** or the **-u** option. For more information on this variable, see **timezone(3)**.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- >0 An error occurred.

Messages

- Message:** bad date conversion in "*string*"
- Cause:** The date and/or time specified on the command line had an invalid format (for example, the hour is greater than 24).
- Action:** Check the *DESCRIPTION* section of this man page for valid date formats.

-
- Message:** bad format or date output longer than *number* bytes
Cause: The format string supplied to **date** is invalid, or the output is longer than *number* bytes where *number* is the value of the configuration variable `LINE_MAX` (see also **getconf**(1)).
Action: Confirm that the date format string on the command line is valid. or modify your date format to produce a shorter output string.
- Message:** no permission to set date
Cause: You do not have proper permissions for changing the system date.
Action: If you need the system date changed, talk to your system manager.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for this command.
Action: Check the *DESCRIPTIONS* section for a list of valid options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems. The **-c** option is an extension to the POSIX standard.

MPE/iX NOTES

The current MPE/iX implementation of **date** does not allow you to set the date and time.

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

touch(1), **timezone**(3)

NAME

dc — arbitrary precision desk calculator

SYNOPSIS

dc [*file*]

DESCRIPTION

dc is a desk calculator program that takes input in reverse Polish notation (see *Reverse Polish Notation* later in this man page). If you do not specify a file on the command line, **dc** reads input from the standard input; otherwise, it reads input from the file and then from the standard input (if there is no quit command in the file). **dc** sends output to the standard output.

There are several types of input:

- (a) *Numbers* are sequences of digits, possibly containing a decimal point. Numbers can also contain the uppercase characters A through F standing for the hexadecimal (base 16) digits greater than ten; for more on hexadecimal, see the section on *Numbers in Different Bases*. Do not break up a number with spaces or commas; for example, you must write 1000000, not 1,000,000. To create a negative number, put an underscore (_) immediately before the first digit of the number. Do not use a minus sign (-) to indicate a negative number; the minus sign has an entirely different meaning in **dc** input.
- (b) *Strings* are sequences of characters, enclosed in square brackets. For example, [abc] is a string which contains the characters abc.
- (c) *Operators* are symbols or characters telling **dc** to perform some operation; for example, adding two numbers together.
- (d) *Register names* are single characters. You may use any character as a register name. An uppercase letter is not the same as the corresponding lowercase one, so register a is different from register A. A register is a place where **dc** can store a number or a string; it is similar to a variable in a programming language. Typically, you use registers to store values that you want to remember for later use.
- (e) *Array names* follow the same rules as register names. See *Array Operations* later in this man page for more details.

You must separate adjacent numbers with at least one white space character. (The white space characters for **dc** are the blank, the horizontal tab, and the newline.) You do not need to separate other pieces of input from one another, but putting in white space characters makes the input more readable. As exceptions, register names and array names must immediately follow the operator that tells what you want to do with the register or array (as described later in this man page). If you put a white space character after an operator that expects a register or array name, **dc** assumes the white space character to be the name.

Reverse Polish Notation

To use **dc** you must understand reverse Polish notation. This is a way to write arithmetic expressions. The form is a bit tricky for people to understand, since it is geared towards making it easy for the computer to perform calculations; however, most people can get used to the notation with a bit of practice.

Reverse Polish notation stores values in a *stack*. A stack of values is just like a stack of books: one value is placed on top of another. When you want to perform a calculation, the calculation uses the top numbers on the stack.

For example, here's a typical addition operation:

```
1 2 +
```

When **dc** reads a number or a string, it just puts the value onto the stack. Thus 1 goes on the stack, then 2 goes on the stack. When you put a value onto the stack, we say that you *push* it onto the stack. When **dc** reads the operator +, it takes the top two values off the stack, adds them, then pushes the result back onto the stack. After this addition, the stack contains

```
3
```

As another example, consider

```
2 3 4 + *
```

(The * stands for multiplication.) **dc** begins by pushing the three numbers onto the stack. When it finds the +, it takes the top two numbers off the stack and adds them. (Taking a value off the stack is called *poping* the stack.) **dc** then pushes the result of the addition back onto the stack in place of the two numbers. Thus the stack contains

```
2 7
```

When **dc** finds the * operator, it again pops the top two values off the stack. It multiplies them, then pushes the result back onto the stack, leaving

```
14
```

The following list gives a few more examples of reverse Polish expressions. After each, we show the contents of the stack, in parentheses.

```
7 2 -          (5)
2 7 -          (-5)
12 3 /         (4)
_12 3 /        (-4)
```

```

4 5 + 2 *      (18)
4 5 2 + *      (28)
4 5 2 * -      (-6)

```

If you are experimenting with **dc** to see how this works, you can type **p** to print out the top value on the stack and **f** to print out the full stack.

The Scaling Factor

One of **dc**'s great virtues is its ability to deal with numbers of arbitrary size and precision — **dc** is not constrained by the hardware's restrictions on number size or precision.

Many arithmetic calculations use a *scaling factor*, an integer greater than or equal to zero, and strictly less than 100. The scaling factor affects how many decimal places **dc** uses when making calculations.

The default scaling factor begins at zero (no decimal places). This can be confusing; for example, if you try

```
1 2 / p
```

to divide 1 by 2 and print the result, **dc** prints 0. The real answer is 0.5 but a scaling factor of 0 tells **dc** not to keep track of fractions when doing arithmetic.

You can set a different default scaling factor with the **k** operation. This pops the top value from the stack and sets that value to the new default scaling factor. For example,

```
4 k
```

sets the default scaling factor to 4. Now if you try

```
1 2 / p
```

the result is .5000.

As our discussion of arithmetic operations points out, the number of decimal places in the operands also affects the number of decimal places in the answer. Thus the scaling factor is not the only influence on the precision of the calculations.

The **K** operation pushes the current default scaling factor onto the stack.

Basic Operators

The following list is the operators recognized by **dc** and the effects that they have.

- +** pops the top two values from the stack, adds them, then pushes the result onto the stack. The number of decimal places in the result is the maximum number of decimal places in the two operands; the scaling factor has no effect.

-
- pops the top two values from the stack, subtracts the first popped from the second, then pushes the result onto the stack. The number of decimal places in the result is the maximum number of decimal places in the two operands; the scaling factor has no effect.
 - * pops the top two values from the stack, multiplies them, then pushes the result onto the stack. **dc** normally sets the number of decimal places in the result to the sum of the decimal places in the two operands; if this is larger than the scaling factor and also larger than the number of decimal places in both individual operands, the number of decimal places in the result is the largest of the scaling factor or the number of decimal places in either operand.
 - / pops the top two values from the stack, divides the second popped by the first, then pushes the result onto the stack. The number of decimal places in the result is equal to the scaling factor.
 - % pops the first two values from the stack, divides the second popped by the first, then pushes the *remainder* onto the stack. (Mathematically, $A \ B \%$ calculates A modulo B.) **dc** determines the number of decimal places in the result by the result of the division.
 - ^ pops the first two values from the stack, calculates the second popped to the power of the first, then pushes the result onto the stack. For example,


```
2 3 ^
```

 leaves the value 8 on the stack. The exponent value must be an integer (that is, with no decimal places). The scaling factor of the result is the scaling factor you get if the base was multiplied the appropriate number of times.
 - c clears the stack (that is, pops all the values off and discards them).
 - d duplicates the value on top of the stack. For example,


```
d *
```

 duplicates the top value, then does a multiplication. The result is that you square the value on top of the stack. As another example, you can use **d** to save the value on top of the stack in a register while keeping a copy of the value on the stack; in this case, you'd use **d** to duplicate the top value, then use **s** to pop the duplicate value into a register.
 - f prints all values on the stack, from top to bottom. (It does not print the contents of the registers.)
 - K pushes the current default scaling factor onto the stack.

-
- k** pops the top value off the stack and uses it as the default scaling factor (see the section on *The Scaling Factor*).
- Lx** pops the top value off the register stack *x* (see the **S** command) and pushes that value onto the main stack. If the register has never contained a value, **dc** treats this as an error. (Contrast this behavior with the way that the **lx** operator works.) This operator also pops the array component of the specified register. See the *Array Operations* section for more information.
- lx** takes the value from register *x* and pushes it onto the stack. This does not change the value of the register. If the register has never contained a value, **dc** puts a value of zero on the stack.
- P** pops the top value off the stack, prints it as a string, and then discards it. If the value is a string, **dc** prints it as such. If it is a number, **dc** prints the ASCII character with that value.
- p** prints the top value on the stack. The value remains on the stack.
- q** quits a **dc** session; however, see the *Executing Strings* section for an exception.
- Sx** pops the top value off the stack and pushes this value onto the register *x* as if the register itself were another stack. In this way, you can use a single register to hold a sequence of values. This operation also pushes the array component of the register onto the register's stack. See the *Array Operations* section for more information.
- sx** pops the top value off the stack and stores it in the register *x*. For example, **sa** pops the stack and stores the value in register **a**.
- v** replaces the top value on the stack with its square root. **dc** ignores the scaling factor when performing calculations to find the square root. The number of decimal places in the result is the maximum of the number of decimal places in the original value or the scaling factor.
- X** replaces the value on the top of the stack with the number of decimal places in the number.
- x** executes a string. See *Executing Strings*.
- Z** replaces the number on the top of the stack with its length (that is, the number of digits in the number).
- Note:** **dc** ignores the minus sign and decimal point when calculating this value, so that **12345** and **_123.45** have the same length.
- z** determines how many values are currently on the stack, then pushes that number onto the stack.

Numbers in Different Bases

Programmers often find it useful to perform arithmetic with numbers in bases other than ten, for example, octal (base 8) or hexadecimal (base 16) numbers. Several commands help make this possible.

I pushes the current input base onto the stack.

i pops the top value of the stack and uses this as the base when interpreting further input. For example,

```
8i
```

tells **dc** that from now on, it is to interpret input numbers as octal values. For example, if you type 10 as input, **dc** interprets it as an octal number, equal to 8 (base ten).

Note: You can use the characters A through F to input hexadecimal digits regardless of the base.

O pushes the current output base onto the stack.

o pops the top value of the stack and uses this as the base when printing output. For example,

```
16o
```

tells **dc** to print subsequent numbers in hexadecimal format.

Note: The input and output bases can be different; for example, you may find this convenient if you want to convert input in one base to output in another.

You can make the output base larger than 16. In this case, **dc** prints each *digit* as a decimal value and separates them with a single space. For example,

```
1000 o
123456789 p
```

prints

```
123 456 789
```

This sets the output base to 1000, where *digits* are decimal values from 0 through 999. As a result, **dc** breaks up all values into one or more *chunks* with three digits per chunk. Using output bases that are large powers of ten, you can put your output in columns; for example, many users find that 100000 makes a good output base because **dc** groups numbers into chunks of five digits each.

dc outputs long numbers with a maximum of 70 characters per line. If a number is longer than this, **dc** puts a backslash \ at the end of the line, indicating that the number continues on the next line.

dc always prints a value of zero as 0, regardless of the output base and regardless of the number of decimal places that are normally attached to the value.

Some people have trouble figuring out how to put the input base back to base ten after working in some other base.

```
A i
```

always works, since A stands for the hexadecimal digit ten.

The maximum output base is the maximum integer value that the hardware can represent.

Executing Strings

A string is any sequence of characters. In particular, a string may consist of a sequence of **dc** commands.

The **x** command pops the top value from the stack and executes it as if it were a string containing **dc** commands. For example, consider the following code:

```
[lapP lbpP]sz
```

This pushes the string inside the square brackets onto the stack, and then pops it into register **z**. From this point onward, the command

```
lzx
```

pushes the string in **z** onto the stack, then execute the commands inside the string. The sequence of commands **lapP** pushes the value of register **a** onto the stack, prints it, then pops the value off again. The sequence of commands **lbpP** does the same for register **b**. The result is that we can use **lzx** to print the current contents of registers **a** and **b** any time we want.

There are several other commands for executing strings:

>x pops two values off the stack. If the first popped value is greater than the second, **dc** executes the contents of register **x** as a string of commands. As an example,

```
la lb >z
```

executes the string in register **z** if the contents of register **b** are greater than the contents of register **a**.

!>x pops two values off the stack. If the first popped value is not greater than the second, **dc** executes the contents of register **x** as a string of commands.

-
- <*x* pops two values off the stack. If the first popped value is less than the second, **dc** executes the contents of register *x* as a string of commands.
 - !<*x* pops two values off the stack. If the first popped value is not less than the second, **dc** executes the contents of register *x* as a string of commands.
 - =*x* pops two values off the stack. If the first popped value is equal to the second, **dc** executes the contents of register *x* as a string of commands.
 - !=*x* pops two values off the stack. If these two values are not equal, **dc** executes the contents of register *x* as a string of commands.

One string may execute another. For example, a string being executed via the *x* command may contain a *>* construction to execute a register string if the condition holds true. In this case, **dc** executes the new string, then returns to the old string to continue executing where it left off. A string may execute a string which executes another string, and so on. Because of this possibility, **dc** keeps a *stack* of the strings that it is currently executing.

When **dc** finds a *q* command inside a string being executed, it doesn't quit **dc**. Instead, it quits executing the current string, plus the string that caused the execution of the current string. In other words, it pops two strings off the currently executing stack.

To see why you want to quit the *two* most recent strings, consider the following example.

```
[q]sy
```

loads a quit command into register *y*. Now, we might use something like

```
[... la lb >y ...]
```

to quit in the middle of the string if the value in register *b* is greater than the value in register *a*. The command *>y* executes the command string in register *y* if the condition is true; if the quit command in *y* only stopped one command string, it would quit executing the commands from *y* and go right back to executing the main command string. To be able to use this technique to quit the main command string, the *q* command must pop *two* command strings.

The *Q* (uppercase) command is a variation on the simple *q* command. *Q* pops the top number off the (value) stack and stops execution of that many currently executing strings. For example,

```
3Q
```

stops the three most recent executing strings.

Array Operations

As noted previously, arrays are similar to registers in that they have names consisting of a single character. However, a register's array values are independent of its scalar value; the array element *X[1]* for example, is different than the scalar *X*.

An array is just a list of values. Values in the list are referred to by number; for example, you can ask for the 12th value in the list. The numbers used to refer to values are called the *subscripts* of the array. The beginning of the list has the subscript 0 and the maximum subscript for any array is 2047.

There are two array operations:

- `:x` stores a value in array *x*. The operation begins by popping a number off the stack and uses it as the subscript into the array. **dc** then pops another value off the stack and stores this value in the array using the given subscript.
- `;x` obtains a value from the array *x*. **dc** pop the top number off the stack to use as a subscript into the array. It then places the value found at that subscript on the stack. The operation does not affect the value inside the array; it just takes a copy of the value.

If you use `;` to obtain a value from an array, but you have not yet used `:` to store a value in that position, **dc** automatically puts a zero onto the stack (as if there were a zero in that position).

In an earlier section, the `S` and `L` operators were used to push and pop the scalar value of a register onto the register's stack. These operators also push and pop the array component of a register. This is done at the same time that the scalar values are being pushed or popped with some differences in the details of how operations work. Where `L` popped the top of the register's scalar stack onto the main stack, the array operation simply pops the top of the register's array stack then discards the result. Where `S` popped the top of the main stack and pushed it onto the register's scalar stack, the array operation simply hides the current array values. Again, both the scalar and array operations are caused by the same operator at the same time. The following example shows how the `S` and `L` operations can be used to save or hide the scalar and array values of a register. The operations

```
11 sa 12 1 :a la p 1 ;a p c
```

store 11 in `a` and 12 in `a[1]` then print the two values and finally clear the main stack. The register `a` now has the scalar value 11 and the array element `a[1]` now has the value 12. Next, the operations

```
0 Sa la p 1 ;a p
```

save the current array and scalar values associated with the register and print the new values for `a` and `a[1]` (which are now zero). The old array and scalar values of the register have been saved on the register's stack. You can change the value of the register or any of the array elements without affecting these saved values. To restore the old values, execute

```
La la p 1; a p
```

This pops the current array and scalar values off of the register stack thus making the old values visible again. The restored values are then printed.

Other Commands

!command

executes the rest of the line as a system command. For example,

```
!cp file1 file2
```

executes the given **cp** command.

? reads an input line from the input source (for example, the terminal) and executes that line. This is useful when you are executing a command string but want to obtain input in the middle of the string.

EXAMPLE

The following sequence of commands prints out the first 12 elements of the Fibonacci sequence. In this sequence, the first two values are 1, and each subsequent value is the sum of the previous two values. Registers **a** and **b** hold the two most recent values of the sequence; new values are calculated in the stack. Register **z** holds the code needed to calculate new values, and register **c** holds a count of how many values have been printed.

```
1 sa
1 sb
2 sc
[la lb + p lb sa sb lc 1 + d sc 13 >z] sz
la p sx lp p sx lz x
```

The first three lines set up registers **a** and **b** with the value 1, and **c** with the count of values that have already been calculated (the first two). The next line loads **z** with the main code to execute. This code loads the values in **a** and **b**, adds them and prints the result, moves the value of **b** to **a**, then saves the newly calculated value in **b**. The count in **c** is then incremented; the commands

```
d sc
```

make a duplicate copy of the count and save this duplicate back in **c**. The final part of the code checks the count to see if it is less than 13; if this is true, the contents of **z** are executed again to get the next value. The final line in the program prints the first two values of the sequence and then executes the code in **z**.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** can not execute number
Cause: You attempted to use the `x` operator to execute a string, but the value on the top of the stack was a number.
Action: Only use the `x` operator when there is a string on top of the stack.
- Message:** command too long
Cause: You specified a command line to pass to the system with the `!` operator that was longer than 1000 bytes.
Action: Use a shorter command line.
- Message:** divide by 0
Cause: You attempted to divide by 0.
Action: Do not divide by zero.
- Message:** empty stack
Cause: You attempted an operation that required popping a value from the stack, but the stack was empty.
Action: Push a value onto the stack and try the operation again.
- Message:** exponent must be an integer from 0 to *max*
Cause: When using the `^` operator, the second value popped from the stack (the exponent) was not an integer or was not in the range 0 to *max*.
Action: Make sure that the value which is to be used as the exponent is an integer in the range 0 to *max*.
- Message:** *filename* : system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** index too big
Cause: You attempted to use an array index that was greater than 2047.
Action: Use an array index that is less than or equal to 2047.
- Message:** input radix too big
Cause: You specified an input radix (base) that is too large for **dc** to handle.
Action: Specify a smaller input radix.

-
- Message:** negative argument to Q
Cause: You attempted to use the Q operator but the value on the top of the stack was negative. Q cannot take a negative argument.
Action: Make sure that the stack has a positive number on top when using the Q operator.
- Message:** negative index
Cause: You attempted to use a negative number as an array index.
Action: Use a positive number as an array index.
- Message:** number expected, string found
Cause: When popping a value from the stack, **dc** expected a number but a string value was found.
Action: Make sure that the stack contains the proper type of value when performing operations using the stack.
- Message:** numerical constant is too long
Cause: You specified a numerical constant that had more than 1000 digits.
Action: Use a numerical constant with less than 1000 digits.
- Message:** *oct_num* is unimplemented
Cause: You specified a character which is not a currently implemented operators. *oct_num* is the octal value of the character.
Action: See the *DESCRIPTION* section of this man page for a list of valid **dc** operators.
- Message:** out of memory (fatal)
Cause: **dc** ran out of system resources.
Action: Free up some resources, check for infinite recursion when executing strings, and make sure information is not being left on the stack.
- Message:** output radix too big
Cause: You specified an output radix that is too large for **dc** to handle.
Action: Specify a smaller output radix.
- Message:** readstk?
Cause: You attempted to pop too many values off the stack with the Q operator.
Action: Make sure that the top value on the stack is not greater than the number of currently executing strings.
- Message:** save: args
Cause: You attempted to use the s or S operator when there was no value on the stack.
Action: Make sure that there is at least one value on the stack before trying to use the s and S commands.

Message: scale too big
Cause: You specified a scaling factor that was too large for **dc** to handle.
Action: Specify a smaller scaling factor.

Message: sqrt of negative number
Cause: You attempted to take the square root of a negative number.
Action: Only use the $\sqrt{\quad}$ (square root) operator on positive numbers.

Message: stack too deep
Cause: You attempted to put more values on the stack than it was able to hold. The maximum size of the stack is limited by the size of the maximum integer your system can represent.
Action: Check for uncontrolled recursion.

Message: string is too long
Cause: You specified a string that is too long for **dc** to handle.
Action: Split the string into smaller strings.

LIMITS

Maximum array index: 2047.
Maximum exponent in an exponentiation operation: 9999.
Maximum input buffer size (line length): 1000 characters.
Maximum scaling factor: 99.
Maximum stack depth: MAXINT (that is, the size of the largest positive integer that can be supported by the hardware).

PORTABILITY

UNIX System V.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

bc(1)

NAME

dd — copy and convert input blocks

SYNOPSIS

```
dd [bs=s] [cbs=s] [conv=conversion] [count=n] [ibs=s] [if=file]
[img=string] [iseek=n] [obs=s] [of=file] [omsg=string] [seek=n]
[skip=n]
```

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

dd reads and writes data by blocks. It is frequently used for devices such as tapes which have discrete block sizes, or for fast multi-sector reads from disks. **dd** performs conversions to accommodate computers that require de-blocking, conversion to/from EBCDIC, and fixed length records.

dd processes the input data as follows:

1. **dd** reads an input block.
2. If this input block is smaller than the specified input block size, **dd** pads it to the specified size with null bytes. When you also specify a block or unblock conversion, **dd** uses spaces instead of null bytes.
3. If you specified **bs=size** and requested no conversion other than **sync** or **noerror**, **dd** writes the padded (if necessary) input block to the output as a single block and omits the remaining steps.
4. If you specified the **swab** conversion, **dd** swaps each pair of input bytes. If there is an odd number of input bytes, **dd** does not attempt to swap the last byte.
5. **dd** performs all remaining conversions on the input data independently of the input block boundaries. A fixed-length input or output record may span these boundaries.
6. **dd** gathers the converted data into output blocks of the specified size. When **dd** reaches the end of the input, it writes the remaining output as a block (without padding if **conv=sync** is not specified). As a result, the final output block may be shorter than the output block size.

bs=size

sets both input and output block sizes to *size* bytes. You can suffix this decimal number with *w*, *b*, *k*, or *xnumber*, to multiply it by 2, 512, 1024, or *number* respectively. You can also specify *size* as two decimal numbers (with or without suffixes) separated by *x* to indicate the product of the two values. Processing is faster when

ibs and **obs** are equal, since this avoids buffer copying. The default block size is 1b. **bs=size** supercedes any settings of **ibs=size** or **obs=size**.

If you specify **bs=size** and you request no other conversions than **noerror**, **notrunc**, or **sync**, **dd** writes the data from each input block as a separate output block; if the input data is less than a full block and you did not request **sync** conversion, the output block is the same size as the input block.

cbs=size

sets the size of the conversion buffer used by various **conv** options.

conv=conversion[, conversion, ...]

where *conversion* can be any of the following:

ascii	converts EBCDIC input to ASCII for output. dd copies cbs bytes at a time to the conversion buffer, maps them to ASCII; then strips trailing blanks, adds a newline, and copies this line to the output buffer.
block	converts variable-length records to fixed-length records. dd treats the input data as a sequence of variable-length records (each terminated by a newline or an EOF character) independent of the block boundaries. dd converts each input record by first removing any newline characters, then padding (with spaces) or truncating the record to the size of the conversion buffer. dd reports the number of truncated records on the standard error. You must specify cbs=size with this conversion.
ebcdic	converts ASCII input to EBCDIC for output. dd copies a line of ASCII to the conversion buffer, discards the newline, pads it with trailing blanks to cbs bytes, maps it to EBCDIC and copies it to the output buffer.
ibm	converts ASCII to a variant of EBCDIC which gives better output on many IBM printers.
lcase	converts uppercase input to lowercase.
noerror	ignores errors on input.
notrunc	does not truncate the output file. dd preserves blocks in the output file to which it does not write explicitly write.
swab	swaps the order of every pair of input bytes. If the current input record has an odd number of bytes, this conversion does not attempt to swap the last byte of the record.

<code>sync</code>	pads any input block shorter than ibs to that size with null bytes before conversion and output. If you also specified <code>block</code> or <code>unblock</code> , dd uses spaces instead of null bytes for padding.
<code>ucase</code>	converts lowercase input to uppercase.
<code>unblock</code>	converts fixed-length records to variable-length records by reading a number of bytes equal to the size of the conversion buffer, deleting all trailing spaces, and appending a newline character. You must specify cbs=size with this conversion.
<code>convfile</code>	uses <i>convfile</i> as a translation table if it is not one of the conversion formats listed here and it is the name of a file of exactly 256 bytes.

You may perform multiple conversions at the same time by separating arguments to **conv** with commas; however, some conversions are mutually exclusive (for example, `ucase` and `lcase`).

count=n

copies only *n* input blocks to the output.

ibs=size

sets the input block size to *size* bytes. You specify it in the same way as **bs**.

if=file reads input data from *file*. If you don't specify this option, **dd** reads data from the standard input.

imsg=string

displays *string* when all data has been read from the current volume, replacing all occurrences of `%d` in *string* with the number of the next volume to be read. **dd** then reads and discards a line from the controlling terminal, giving you a chance to change volumes (usually diskettes).

iseek=n

seeks to the *n*th block of the input file. The distinction between this and **skip** is that **iseek** does not read the discarded data; however there are some devices, such as tape drives and communication lines, on which seeking is not possible, so only **skip** is appropriate.

obs=size

sets the output block size to *size* bytes. You specify it in the same way as the **bs** value. The size of the destination should be a multiple of the value chosen for *size*. For example, if you choose **obs=10k**, the destination's size should be a multiple of 10k.

of=file writes output data to *file*. If you don't specify this option, **dd** writes data to the standard output. **dd** truncates the output file before writing to it, unless you specified the **seek=n** operand. If you specify **seek=n**, but do not specify **conv=notrunc**, **dd** preserves only those blocks in the output file over which it seeks. If the size of the seek plus the size of the input file is less than the size of the output file, this can result in a shortened output file.

omsg=string

displays *string* when **dd** runs out of room while writing to the current volume. Any occurrences of %d in *string* are replaced with the number of the next volume to be written. **dd** then reads and discards a line from the controlling terminal, giving you a chance to change volumes (usually diskettes).

seek=n

initially seeks to the *n*th block of the output file.

skip=n

reads and discards the first *n* blocks of input.

EXAMPLES

```
dd if=in of=out conv=ascii cbs=80 ibs=6400 obs=512
```

Converts 80-byte fixed length EBCDIC punch card images in 6400 byte input blocks to variable length ASCII lines, 512 bytes to the output block.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - an option that should contain = does not
 - I/O errors on read/write
 - invalid command line option

Messages

Message: absolute I/O must be in *num* byte units

Cause: You attempted to read from, or write to, a device which requires block sizes to be in multiples of its sector size (in this case, *num* bytes).

Action: Specify a block size that is a multiple of the device's sector size.

-
- Message:** badly formed number "*num*"
Cause: You specified *num* as a number (for example, a block size), but *num* did not have the form of a number recognized by **dd**.
Action: Make sure that *num* is a valid number, and if it is followed by a letter to indicate the block size unit, check the *DESCRIPTION* section of this man page under the **bs=** option for a list of valid letters.
- Message:** *cbs=size* given without *ascii/ebcdic/ibm/block/ unblock conversion*
Cause: You specified the **cbs=size** option but did not specify a conversion option which uses it.
Action: Provide the missing conversion option.
- Message:** *conv=block/unblock* given without *cbs=* option
Cause: You specified either the **conv=block** or **conv=unblock** option without defining a block size with the **cbs=size** option.
Action: Provide the missing **cbs=size** option.
- Message:** input file "*filename*"
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** *option=value* is an unknown option
Cause: You specified an option that is not valid for **dd**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **dd** man pages.
- Message:** out of memory for buffers
Cause: **dd** was unable to allocate the system resources that it needed for conversion buffers.
Action: Free up more system resources.
- Message:** output file "*filename*"
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** read error: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** seek input: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

Message: seek output: *system error*

Cause: See **syserror**(3).

Action: See **syserror**(3).

Message: The option "*option*" does not contain a "="

Cause: You specified *option* without providing the required equals sign (=).

Action: Provide the missing equals sign.

Message: unknown conversion "*conv*"

Cause: You specified a conversion value following **conv=** that **dd** did not recognize.

Action: Check the *DESCRIPTION* section of this man page for a list of valid conversion values.

Message: write error: *system error*

Cause: See **syserror**(3).

Action: See **syserror**(3).

PORTABILITY

POSIX.2. *x/OPEN* Portability Guide 4.0. All UNIX systems.

The **conv=ascii**, **conv=ebcdic**, **conv=ibm**, **conv=convfile**, **iseek**, **img**, and **msg** options plus the **w** suffix described in the **bs=** option are all extensions to the POSIX standard.

MPE/iX NOTES

The current MPE/iX implementation of **dd** converts non-byte stream files to byte stream files before processing them. File characteristics like file code, record size, and so forth are not preserved by this conversion. The output of **dd** is written as a byte stream file.

In addition, direct device input/output is not currently implemented.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

cp(1), **cpio**(1), **mv**(1), **tr**(1)

NAME

diff, diffh, bdiff — compare two text files and show differences

SYNOPSIS

```
diff [-befHhimnrstw] [-C n] [-c[n]] [-Difname] path1 path2
diffh [-befimnrstw] [-C n] [-c[n]] [-Difname] path1 path2
bdiff [-befimnrstw] [-C n] [-c[n]] [-Difname] path1 path2 [n]
```

DESCRIPTION

The **diff** command attempts to determine the minimal set of changes needed to convert a file named *path1* into *path2*.

If either (but only one) file name is **-**, **diff** reads from standard input. If exactly one of *path1* or *path2* is a directory, **diff** uses a file in that directory with the same name as the other file name. If both are directories, **diff** compares files with the same file names under the two directories; however, it does not compare files in subdirectories unless you specify the **-r** option. When comparing two directories, **diff** does not compare block special files, character special files, or FIFO special files to any other files and does not compare regular files to directories.

By default, output consists of descriptions of the changes in a style reminiscent of the **ed** text editor. A line indicating the type of change is given. The three types are **a** (append), **d** (delete), and **c** (change). The output is symmetric in the sense that a delete in *path1* is the counterpart of an append in *path2*. **diff** prefixes each operation with a line number (or range) in *path1* and suffixes each with a line number (or range) in *path2*. After the line giving the type of change, **diff** displays the deleted or added lines, prefixing lines from *path1* with **<** and lines from *path2* with **>**.

When you call the command as **diffh**, it automatically uses the **-h** option.

When you call it as **bdiff**, **diff** computes the differences in chunks of *n* lines (default 3999). This lets you process arbitrarily large files and generally produces less output than the **-h** option.

Options

diff accepts the following options:

- b** ignores white space preceding the newline at the end of each line and considers strings of white space elsewhere in input lines to be equivalent. For example if one file contained a string of three spaces and a tab at a given location while the other file contained a string of two spaces at the same location, **diff** would not report this as a difference.

-
- C *n*** is equivalent to **-cn**.
 - c[*n*]** shows *n* lines of context before and after each change. The default value for *n* is 3. **diff** marks lines removed from *path1* with -, lines added to *path2* with + and lines changed in both files with !.
 - D*ifname***
displays output that is the appropriate input to the C preprocessor to generate the contents of *path2* when *ifname* is defined, and the contents of *path1* when *ifname* is not defined.
 - e** writes out a script of commands for the **ed** text editor, which converts *path1* to *path2*. **diff** sends the output to the standard output.
 - f** writes a script similar to the one produced under **-e** to the standard output, but does not adjust the line numbers to reflect earlier editing changes; instead, they correspond to the line numbers in *path1*.
 - H** uses the half-hearted (**-h**) algorithm only if the normal algorithm runs out of system resources.
 - h** uses a fast, half-hearted algorithm instead of the normal **diff** algorithm. This algorithm can handle arbitrarily large files; however, it is not particularly good at finding a minimal set of differences in files with many differences.
 - i** ignores the case of letters when doing the comparison.
 - m** produces the contents of *path2* with extra formatter request lines interspersed to show which lines were added (those with vertical line characters in the right margin) and deleted (indicated by a * in the right margin). These are **nroff/troff** requests.
 - n** displays the differences in a form that is usable by RCS.
 - r** compares corresponding files under the directories, and recursively compares corresponding files under corresponding subdirectories under the directories. You can use this option when you specify two directory names on the command line.
 - s** compares two directories, file by file, and prints messages for identical files between the two directories.
 - t** expands tabs into spaces before doing the comparison. **diff** sets tab stops every eight columns, for example columns 1, 9, 17.
 - w** ignores white space when making the comparison.

EXAMPLES

The following example illustrates the effect of the `-c` option on the output of the `diff` command. The following two files, `price1` and `price2`, are compared with and without the use of the `-c` option.

The contents of `price1` are as follows:

```
Company X Price List:

$ 0.39  -- Package of Groat Clusters
$ 5.00  -- Candy Apple Sampler Pack
$ 12.00 -- Box of Crunchy Frog Chocolates
$ 15.99 -- Instant Rain (Just Add Water)
$ 20.00 -- Asparagus Firmness Meter
$ 25.00 -- Package of Seeds for 35 Herbs
$ 30.00 -- Child's Riding Hood (Red)
$ 35.00 -- Genuine Placebos
$ 45.00 -- Case of Simulated Soy Bean Oil
$ 75.88 -- No-Name Contact Lenses
$ 99.99 -- Kiddie Destructo-Bot
$125.00 -- Emperor's New Clothes
```

The contents of `price2` are as follows:

```
Company X Price List:

$ 0.39  -- Package of Groat Clusters
$ 5.49  -- Candy Apple Sampler Pack
$ 12.00 -- Box of Crunchy Frog Chocolates
$ 15.99 -- Instant Rain (Just Add Water)
$ 17.00 -- Simulated Naugahyde cleaner
$ 20.00 -- Asparagus Firmness Meter
$ 25.00 -- Package of Seeds for 35 Herbs
$ 30.00 -- Child's Riding Hood (Red)
$ 35.00 -- Genuine Placebos
$ 45.00 -- Case of Simulated Soy Bean Oil
$ 75.88 -- No-Name Contact Lenses
$ 99.99 -- Kiddie Destructo-Bot
```

The command

```
diff price1 price2
```

results in the following output:


```

4c4
< $ 5.00 -- Candy Apple Sampler Pack
---
> $ 5.49 -- Candy Apple Sampler Pack
6a7
> $ 17.00 -- Simulated Naugahyde cleaner
14d14
< $125.00 -- Emperor's New Clothes

```

The addition of the `-c` option, as in

```
diff -c price1 price2
```

results in the following output:

```

*** price1 Wed Mar 04 10:08:40 1992
--- price2 Wed Mar 04 10:09:10 1992
*****
*** 1,9 ****
    Company X Price List:

    $ 0.39 -- Package of Groat Clusters
! $ 5.00 -- Candy Apple Sampler Pack
    $ 12.00 -- Box of Crunchy Frog Chocolates
    $ 15.99 -- Instant Rain (Just Add Water)
    $ 20.00 -- Asparagus Firmness Meter
    $ 25.00 -- Package of Seeds for 35 Herbs
    $ 30.00 -- Child's Riding Hood (Red)
--- 1,10 ----
    Company X Price List:

    $ 0.39 -- Package of Groat Clusters
! $ 5.49 -- Candy Apple Sampler Pack
    $ 12.00 -- Box of Crunchy Frog Chocolates
    $ 15.99 -- Instant Rain (Just Add Water)
+ $ 17.00 -- Simulated Naugahyde cleaner
    $ 20.00 -- Asparagus Firmness Meter
    $ 25.00 -- Package of Seeds for 35 Herbs
    $ 30.00 -- Child's Riding Hood (Red)
*****
*** 11,14 ****
    $ 45.00 -- Case of Simulated Soy Bean Oil
    $ 75.88 -- No-Name Contact Lenses

```

```

    $ 99.99  -- Kiddie Destructo-Bot
-   $125.00 -- Emperor's New Clothes
--- 12,14 ----

```

diff -c marks lines removed from `price1` with `-`, lines added to `price1` with `+` and lines changed in both files with `!`. In the example, **diff** shows the default 3 lines of context around each changed line. One line was changed in both files (marked with `!`), one line was added to `price1` (marked with `+`), and one line was removed from `price1` (marked with `-`).

Note: If there are no marks to be shown in the corresponding lines of the file being compared, the lines are not displayed. Lines 12 to 14 of `price2` are suppressed for this reason.

DIAGNOSTICS

Possible exit status values are:

- 0 No differences between the files compared.
- 1 **diff** compared the files and found them to be different.
- 2 An error occurred.
- 4 At least one of the files is a binary file, containing embedded NUL (`\0`) bytes or new-lines which are more than `LINE_MAX` characters apart.

Messages

Message: cannot open directory "*pathname*"
Cause: You do not have read permission on *pathname*.
Action: Use **chmod** to acquire read permission on *pathname*.

Message: File *file1* is a *type1* while file *file2* is a *type2*
Cause: The two files specified on the directory were of different file types.
diff can only compare files that have the same file type.
Action: Specify two files that have the same file type.

Message: file "*filename*" is binary
Cause: You specified the binary file *filename* as a **diff** input file. **diff** only works on text files.
Action: Only specify text files as **diff** input files.

Message: file "*filename*": line too long: limit *num*
Cause: The input line is too long.
Action: Try again with a shorter input line.

-
- Message:** *"filename" : system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** files too large, trying "-h" option ...
Cause: You specified the **-H** option, but there were not enough free system resources to handle the files. **diff** will now try to compare the files using the **-h** option.
Action: If you are comparing these two files again, specify the **-h** option on the command line for faster operation.
- Message:** insufficient memory
Cause: There were not enough free system resources for **diff** to run, even if it used the **-h** option.
Action: Free up more system resources.
- Message:** insufficient memory (try diff -h)
Cause: **diff** ran out of system resources when generating the data structures used in the differencing algorithm (see *LIMITS*).
Action: The **-h** option of **diff** requires fewer system resources than the regular **diff** algorithm. This may allow it to succeed where the regular algorithm fails.
- Message:** internal error--cannot create temporary file
Cause: **diff** was unable to create a working file that it needed.
Action: Ensure that you either have a `/tmp` directory or that the environment contains a variable `TMPDIR` which names a directory where **diff** can store temporary files. Also, ensure that you have sufficient permissions on this directory to create a temporary file.
- Message:** Missing #ifdef symbol after -D
Cause: You did not specify a conditional label on the command line after the **-D** option.
Action: Provide a conditional label with the **-D** option.
- Message:** Missing number after "option" option
Cause: You specified *option* but did not specify a number following it.
Action: Specify a number following the *option* option.
- Message:** only one file may be "-"
Cause: Only one of the two files being compared may be the standard input.
Action: Specify `-` (standard input) as, at most, one of the two files to be compared.
- Message:** read error on file *"filename" : system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: strcoll error, cannot malloc space.
Cause: There are not enough free system resources to allocate string space.
Action: Free up more resources.

Message: too many lines in file "*filename*"
Cause: The file *filename* contained more than the value of the configuration variable INT_MAX. Without the **-h** option, **diff** cannot handle a file that large.
Action: Use the **-h** option.

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **diff**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **diff** options.

Message: unlink temp file: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

Message: write error on standard output: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

Message: write error on temporary file: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

LIMITS

The longest input line is 1024 bytes. Except under **bdiff** and **-h**, files are limited to INT_MAX lines.

PORTABILITY

POSIX.2. *x*OPEN Portability Guide 4.0. All UNIX systems.

The **-D**, **-f**, **-H**, **-h**, **-i**, **-m**, **-n**, **-s**, **-t**, and **-w** options; the *n* argument to the **-c** option; and the **diffh** and **bdiff** versions of the command are all extensions to the POSIX standard.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

J. W. Hunt, M. D. McIlroy. *An Algorithm for Differential File Comparison* in *Computing Science Technical Report 41*. Bell Telephone Laboratories.

cmp(1), **comm(1)**, **diff3(1)**, **patch(1)**

NAME

diff3 — compare three text files

SYNOPSIS

```
diff3 [-EeHhXx3] file1 file2 file3 [mark1{mark3}]
```

DESCRIPTION

The **diff3** command compares three versions of a text file found in *file1*, *file2*, and *file3* on a line by line basis using the **diff**(1) command.

diff3 marks ranges of disagreeing text lines with one of the following headers:

```
====1   only file1 is different.  
====2   only file2 is different.  
====3   only file3 is different.  
====    there are differences in all three files.
```

After the header, label lines of either of the two following forms may appear to show what differences apply to which file:

```
f:m,nc  
f:na
```

The label *f* says which of the three files is being described; it is 1, 2, or 3. The first label form shows a change in file *f* from lines *m* to *n* inclusive, with the actual text following the label line. The second form indicates that the following text is appended after line *n* of file *f*.

Options

All of the following options tell **diff3** to produce an editing script instead of the output described earlier. All scripts consist of commands for the line-oriented text editor **ed**(1), which can then be run on this script to make the indicated changes to *file1*.

- E** is similar to **-e**, except that it also checks for overlapping ranges of lines in the changes. **diff3** highlights these ranges in the output.
- e** produces an editor script of only those changes flagged **====** or **====3**.
- H** tells **diff** to use the **-H** option when called by **diff3**.
- h** tells **diff** to use the **-h** option when called by **diff3**.
- X** is similar to **-x** except that it also checks for overlapping ranges of lines in the changes. **diff3** highlights these ranges in the output.

-x produces an editor script of only those changes where all three files differ.

-3 produces an editor script containing changes which occur only in *file3*.

Under the **-E** and **-X** options, **diff3** highlights overlapping regions as follows:

```
<<<<<<< file1
lines from file1
=====
lines from file3
>>>>>>> file3
```

If you specify the arguments *mark1* and/or *mark3* on the command line, **diff3** uses them as labels in this sort of highlighting instead of the names of *file1* and *file3*.

EXAMPLES

Here are two commands that you can submit to the shell.

```
(diff3 -e file1 file2 file3 ; echo '1,$p') | ed -s file1
```

This simply prints *file1* incorporating the changes between *file2* and *file3*; it does not save those changes.

```
(diff3 -e file1 file2 file3; echo 'wq') | ed -s file1
```

This edits *file1* and saves the changes.

DIAGNOSTICS

Possible exit status values are:

- 0 There were no differences among the three files.
- 1 Some differences were found.
- 2 An error occurred.
- 2+n With the options that check for overlapping differences (**-E** and **-X**), the status indicates that there were *n* overlapping ranges of differences; for example, a status of 3 indicates one overlap.

Messages

Message: cannot find "diff" command
Cause: **diff3** was unable to find the **diff** command.
Action: Make sure that the directory which contains **diff** is included in the setting of the *PATH* environment variable.

-
- Message:** child process: *system error*
Cause: **diff3** was unable to fork to run **diff** due to a system error. See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** "diff" command failed
Cause: For some reason, **diff** returned an unexpected error code and was unable to complete its function.
Action: Contact your system manager.
- Message:** diffnvs3: expecting `---`; got `string`
Cause: **diff3** received unexpected output from **diff**.
Action: Contact your system manager.
- Message:** diffnvs3: bad control line *line*
Cause: **diff3** received unexpected output from **diff**.
Action: Contact your system manager.
- Message:** diffnvs3: unexpected end of file
Cause: **diff3** received unexpected output from **diff**.
Action: Contact your system manager.
- Message:** file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Insufficient memory
Cause: **diff3** was unable to allocate storage for all lines in the input files.
Action: Free up more system resources or break up the files.
- Message:** no temporary files available: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** opening temporary file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **diff3**.
Action: Check the *DESCRIPTION* section for a list of valid **diff3** options.
- Message:** You must specify exactly 3 input files
Cause: You did not specify exactly 3 input files on the command line.
Action: Specify 3 input files.

LIMITS

The longest input line is restricted to 1024 characters.

PORTABILITY

All UNIX systems.

The **-E**, **-H**, **-h**, and **-X** options are extensions to traditional implementations of **diff3**.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

diff(1), **ed(1)**

NAME

diffb — compare binary files and show differences

SYNOPSIS

diffb [-n] [-C n] [-c[n]] *file1 file2*

DESCRIPTION

The **diffb** utility indicates which bytes differ in the binary files *file1* and *file2*. Output consists of descriptions of the changes in a style reminiscent of the **ed** text editor. Each description is headed by a line showing the type of change being performed. This line contains three pieces of information:

- the location of the range of bytes in *file1* affected by the change, represented as an offset from the beginning of the file;
- the type of change, which is one of a (append), d (delete), and c (change);
- the location of the range of bytes in *file2* affected by the change.

After the line giving the type of change, the deleted or added bytes are displayed. Non-printable bytes are represented by an escape sequence consisting of a backslash character (\), followed by the ASCII representation of the byte displayed as a three-digit octal number.

The output of **diffb** resembles the output of **diff**(1), except that differences are ranges of bytes rather than ranges of lines. Each displayed set of bytes from *file1* is prefixed by <, and a < appears at the start of each new line in a set of bytes (that is, after each newline character). Similarly, each group of bytes from *file2* is prefixed by >, and a > appears at the start of each new line in a set of bytes. If bytes from both *file1* and *file2* are being displayed, a line consisting of --- separates the two sets of bytes.

Options

diffb accepts the following options:

- C n This is equivalent to -c n.
- c[n] With each difference, n bytes of context before and after each change are shown. The default value for n is 3.
- n The differences are displayed in a form that is usable by RCS.

DIAGNOSTICS

Possible exit status values are:

- 0 The files were identical.
- 1 The files were compared successfully and found to be different.
- 2 An error occurred.

Messages

Message: input file "*filename*": *system error*

Cause: See **syserror(3)**.

Action: See **syserror(3)**.

Message: missing number after **-C**

Cause: You specified the **-C** option without provide a number as its argument.

Action: Provide the missing number.

Message: option "*-option*" unknown

Cause: You specified an option that is not valid for **diffb**.

Action: Check the *DESCRIPTION* section of this man page for a list of valid **diffb** options.

Message: wrong number of arguments

Cause: You did not specify exactly two files to be compared.

Action: Specify exactly two files to be compared.

For a list of error messages common to all RCS utilities, see **rcserror(3)**.

PORTABILITY

diffb is an extension to traditional implementations of RCS.

LIMITS

The set of differences produced by **diffb** is correct but may not be minimal.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

diff(1), **ed(1)**

NAME

dirname — display directory components of path name

SYNOPSIS

dirname *pathname*

DESCRIPTION

dirname strips off the trailing part of a file name. The result is the path name of the directory that contains the file. This is useful in shell scripts.

Note: **dirname** makes no attempt to validate the path name; for validation, use **pathchk(1)**.

dirname follows these rules:

- If *pathname* is `//`, return it.
- Otherwise, if it is all slashes, return one slash.
- Otherwise, remove all trailing slashes.
- If there are no slashes remaining in *pathname*, return period (`.`).
- Otherwise, remove trailing non-slash characters.
- If the remaining string is `//`, return it.
- Otherwise, remove any trailing slashes.
- If the resulting string is empty, return period (`.`).
- Otherwise, return the resulting string.

EXAMPLES

The command

```
dirname src/lib/printf.c
```

produces

```
src/lib
```

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.

dirname(1)

MPE/iX Shell and Utilities

dirname(1)

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

SEE ALSO

basename(1), **pathchk(1)**

NAME

. (dot) — execute shell file in current environment

SYNOPSIS

. *file* [*argument ...*]

DESCRIPTION

. (dot) executes a shell script in the current environment and then returns. Normally the shell executes a command file in a subshell so that changes to the environment by commands like **cd**, **set**, and **trap** are local to the command file. The . (dot) command circumvents this feature.

If there are slashes in the file name, . (dot) looks for the named file. If there are no slashes . (dot) uses the search *PATH* variable to find *file*. This may surprise some people when they use dot to execute a file under the current directory, but their search rules are not set up to look at the current directory. As a result, the shell doesn't find the shell file. If you have this problem, you can use

```
. ./file
```

This indicates that the shell file you want to run is in the current directory. Also, the file need not be executable, even if it is looked for on the *PATH*. If you specify an argument list *argument ...*, . (dot) sets the positional parameters to this list before execution.

ENVIRONMENT VARIABLES

PATH contains a list of directories that .(dot) searches when attempting to find *file*.

DIAGNOSTICS

Possible exit status values are:

- 1 Returned if the path search fails or *file* is unreadable.

Otherwise, the exit status is the exit status of the last command executed from the script.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh**(1) man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

NOTE

This is a special built-in command of the shell.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

cd(1), **set(1)**, **trap(1)**, **sh(1)**

NAME

du — summarize file space usage

SYNOPSIS

du [-a|-s [-ktx]] [pathname ...]

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

du reports the amount of file space used by the files indicated by the given *pathnames*. If *pathname* is a directory, **du** reports the total amount of file space used by all files in that directory and in each subdirectory in its hierarchy. If you do not specify any *pathname*, **du** assumes the current directory. Files with multiple links are only counted once. On systems supporting symbolic links, only the disk space used by the symbolic link is counted.

du measures file space in 512-byte units.

Options

du accepts the following options:

- a generates a report for all files in *pathname*.
- k displays file sizes in 1024-byte (1K) units.
- s does not display file size totals for subdirectories.
- t displays the total amount of space used by all *pathnames* examined.
- x displays file sizes for only those files contained on the same device as *pathname*.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

Message: directory "*pathname*" : *system error*

Cause: See **syserror(3)**.

Action: See **syserror(3)**.

Message: file "*filename*" : *system error*

Cause: See **syserror(3)**.

Action: See **syserror(3)**.

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **du**.
Action: Check the *DESCRIPTION* section for a list of valid **du** options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

The **-t** option is an extension to the POSIX standard.

MPE/iX NOTES

This release of MPE/iX does not provide the `lstat()` API. As a result, this command cannot return information on the link itself. It attempts to determine when a symbolic link has been referenced, but can only return the information on the target of the link, rather than the link itself.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

find(1), **ls(1)**

NAME

echo — display arguments

SYNOPSIS

echo *argument...*

DESCRIPTION

echo writes its *arguments* to the standard output. **echo** accepts these C-style escape sequences:

<code>\a</code>	bell
<code>\b</code>	backspace
<code>\c</code>	removes any following characters including <code>\n</code> and <code>\r</code>
<code>\f</code>	formfeed
<code>\n</code>	newline
<code>\r</code>	carriage return
<code>\t</code>	horizontal tab
<code>\v</code>	vertical tab
<code>\0num</code>	the byte with the numeric value specified by the zero to three digit octal <i>num</i>
<code>\\</code>	backslash

echo follows the final argument with a newline unless it finds `\c` in the arguments. Arguments are subject to standard argument manipulation.

EXAMPLES

One important use of **echo** is to expand file names on the command line, as in:

```
echo *. [ch]
```

This displays the names of all files with names ending in `.c` or `.h`, typically C source and header files. **echo** displays the names on a single line. If there are no file names in the current directory that end in or `.h`, **echo** simply displays the string `*. [ch]`.

echo is also convenient for passing small amounts of input to other filters:

```
echo 'this is\nreal handy' | banner
```

DIAGNOSTICS

echo always returns the status value:

0 Successful completion.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh(1)** man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x/OPEN* Portability Guide 4.0. UNIX System V.

The POSIX.2 standard does not include the escape sequences, so a strictly conforming application cannot use them. **printf** is suggested as a replacement.

On older UNIX systems, the backslash escape sequences are not available; the **-n** option is equivalent to `\c` embedded in an argument.

NOTE

echo is provided as both an external utility and as a built-in shell utility.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

sh(1)

NAME

ed, red — line-oriented text editor

SYNOPSIS

ed [-**bsx**] [-**p** *prompt*] [*file*]

red [-**bsx**] [-**p** *prompt*] [*file*]

DESCRIPTION

ed is a text editor that lets you manipulate text files interactively. **ed** reads the text of a file into memory and stores it in an area called a *buffer*. Various commands let you edit the text in the buffer. Finally, you can write the contents of the buffer back out to the file, overwriting the old contents of the file.

red is a restricted version of **ed**. It is intended to *protect* the novice user by disallowing the **!** command and the ability to access files found anywhere but the current directory.

Options

Both **ed** and **red** accept the following options:

-b lets you edit larger files by restricting the amount of memory dedicated to paging. This frequently makes **ed** run slower.

-p *prompt*

displays the given *prompt* string prompting you to input a command. By default, **ed** does not usually prompt for command input. See the description of the **P** command for more on command prompting.

-s puts **ed** into a *quiet* mode, in which the **E**, **e**, **r**, and **w** commands do not display file size counts; the **e** and **qe** commands do not check buffer modification; and **ed** does not display **!** after calling the shell to execute a sub-command. This mode is particularly useful when you invoke **ed** from within a shell script.

-x performs an **X** command to handle encrypted files properly. See the description of the **X** command for more details.

- is an obsolete version of **-s**.

If the optional *file* argument is present on the command line, **ed** reads the specified *file* into the editor by simulating an **e file** command.

Addresses

You can prefix commands in **ed** with zero, one, or two addresses. These addresses let you reference single lines or ranges of lines in the buffer. You do not need to specify addresses for certain commands that use default addresses. Consult the description for a particular command.

You can construct each address out of the following components:

- .** The single *dot* character represents the *current line number*. Many commands set the *current line number*. For example, the **e** command sets it to the last line of the new file being edited.
- \$** The dollar sign refers to the last line in the buffer.
- n** The number *n* refers to the *n*th line in the buffer.
- /*regex*/** This searches for a line containing a string that matches the basic regular expression *regex* (see **regex**(3)). The search begins at the line immediately following the current line. It proceeds *forward* through the buffer; if **ed** reaches the end of the buffer without finding a match, it wraps around to the first line of the buffer and continues the search. If **ed** does not find a match, the search ends when it reaches the original current line. If it finds a match, the address **/*regex*/** refers to the first matching line. If you omit *regex*, the last used regular expression becomes the object of the search. You can omit the trailing **/**. Within *regex*, **** represents a literal slash and not the *regex* delimiter.
- ?*regex*?** This similar to the previous address form, except that the search goes *backward* through the buffer. If the search reaches the first line in the buffer without finding a match, **ed** wraps around and continues searching backward from the last line in the buffer. If you omit *regex*, the last used regular expression becomes the object of the search. You can omit the trailing **?**. Within *regex*, **\?** represents a literal question mark and not the *regex* delimiter.
- ^*l*** The address is the line marked with the mark name *l*. The name *l* must be a lower-case letter set by the **k** command.

You can combine these basic addresses with numbers using the + and - operators, with the usual interpretation. Missing left operands default to **.** (dot); missing right operands default to 1. Missing right operands also have a cumulative effect; so an address of **--** refers to the current line number less 2.

You can specify address ranges in the following ways:

- a1,a2*** specifies a range of addresses from address *a1* to address *a2*, inclusive. Omitting *a1* and *a2* (that is, specifying only the comma), is equivalent to the range **1, \$**.
- a1;a2*** is similar to the previous form except that **ed** resets the current line number after calculating *a1*, so that the second address, *a2*, is relative to *a1*. Omitting *a1* and *a2* (that is, specifying only the semicolon), is equivalent to **. ; \$**.

If you specify only *a1* and the command requires both *a1* and *a2*, the command operates as though you specified a range of

a1 ; . *command*

- > is equivalent to . , . +22 (that is, page forward) except that it never attempts to address any line beyond \$.
- < is equivalent to . -22 , . (that is, page backward) except that it never addresses any line before line 1.

Commands

Commands generally take a maximum of zero, one, or two addresses, depending upon the particular command. In the following descriptions, we show commands with their default addresses (that is the addresses used when you don't specify any addresses) in a form that shows the maximum number of legal addresses for the command. The **E**, **e**, **r**, **W**, and **w** commands allow you to specify a *file* argument. For these commands, *file* can be either a path name or a shell escape of the form:

!*command-line*

If you use the ! form, **ed** calls the shell identified by the *SHELL* environment variable to execute *command-line*. For the **E**, **e**, and **r** commands, **ed** reads the standard output of this command line in the same way that it read the contents of a file. For the **W** and **w** commands, the command line treats the addressed lines as standard input.

ed accepts the following commands:

- a** appends text *after* the specified line. Valid addresses range from 0 (text is placed at the beginning of the buffer, before the first line) to \$ (text is placed after the last line of the buffer). **ed** reads lines of text from the terminal until a line consisting solely of an unescaped . (dot) is entered. **ed** sets the current line number to the last line appended.
- .,c** changes the addressed range of lines by deleting the lines and then reading new text in the manner of the **a** or **i** commands.
- .,d** deletes the addressed range of lines. The line after the last line deleted becomes the new current line. If you delete the last line of the buffer, **ed** sets the current line number to the new last line. If no lines remain in the buffer, it sets the current line number to 0.
- E [file]** is similar to the **e** command, but **ed** gives no warning if you have changed the buffer.
- e [file]** replaces the contents of the current buffer with the contents of *file*. If you did not specify *file*, **ed** uses the *remembered* file name, if any. In all cases, the **e** command sets the *remembered* file name to the file that it has just read into the buffer. **ed**

displays a count of the bytes in the file unless it is in *quiet* mode. If you have changed the current buffer since the last time its contents were written out, **ed** displays a warning message and does not execute the command. If you enter the **e** command a second time, **ed** and executes the command.

f [*file*] changes the *remembered* file name to *file*. **ed** displays the new *remembered* file name. If you do not specify *file*, **ed** displays the current *remembered* file name.

1,\$**G**/*regexp*/

is similar to the **g** command except that when **ed** finds a line that matches *regexp*, it displays the line and waits for you to type in the command to be executed. You cannot use the **a**, **c**, **i**, **G**, **g**, **V**, and **v** commands. If you enter **&**, the **G** command re-executes the last command you typed in. If you press ENTER, **G** does not execute any command for that line.

1,\$**g**/*regexp*/command

performs *command* on all lines that contain strings matching the regular expression *regexp*. This command works in two passes. In the first pass, **ed** searches the given range of lines and marks all those that contain strings matching the regular expression *regexp*, while the second pass actually performs *command* on those lines. You cannot use **!**, **G**, **g**, **V**, or **v** as *command*. *command* consists of one or more **ed** commands, the first of which must appear on the same line as the **g** commands. All lines of a multi-line command list, except the last, must end with a backslash (****). If *command* is empty, **ed** assumes it to be the **p** command. If no lines match *regexp*, **ed** does not change the current line number; otherwise, the current line number is the one set by the last command in *command*. You can use any character other than space or newline instead of the slash (**/**) to delimit *regexp*.

H tells **ed** to display more descriptive messages when errors occur. If **ed** is already displaying descriptive messages, because of a previous **H** command, issuing the **H** commands returns to terse error messages. Normally, **ed** indicates error messages by displaying a **?**. When you turn on descriptive error messages with this command, **ed** also displays the descriptive message for the most recent **?** message (see the description of the **h** command).

h provides a brief explanation of the last error that occurred. This does not change the current line number.

i works similarly to the **a** command except that **ed** places the text *before* the addressed line. Valid addresses range from line 1 to **§** (the last line). **ed** sets the current line number to the last inserted line.

..**+1j** joins a range of lines into one line. To be precise, the **j** command removes all newline characters from the addressed range of lines, except for the last one. **ed** sets the current line number to the resulting combined line.

-
- .k***l* marks the addressed line with the mark name *l* which is a single lowercase letter of the alphabet. This lets you refer to a marked line with the construct `'l`. This is called an *absolute address* because it always refers to the same line, regardless of changes to the buffer.
- .,l** displays the addressed range of lines, representing non-printable (control) characters visibly. **ed** sets the current line number to the last line so displayed. You can append this command to most other commands to check on the effect of those commands.
- .,ma** moves the addressed lines to the point immediately following the line given by the address *a*. The address *a* must not be in the range of addressed lines. If address *a* is 0, **ed** moves the lines to the beginning of the buffer. The last line moved becomes the new current line.
- .,n** displays the addressed lines in a way similar to the **p** command, but **ed** puts the line number and a tab character at the beginning of each line. The last line displayed becomes the new current line. You can append **n** to any command other than **E**, **e**, **f**, **Q**, **q**, **r**, **w**, or **!** to check on the effect of that command.
- P** turns on command prompting. If you specified the **-p** *prompt* option on the **ed** command line, **ed** displays the *prompt* string whenever it is ready for you to type in another command. If you did not specify **-p**, **ed** uses the `*` character as a prompt. If command prompting is currently turned on, issuing the **P** command turns it off.
- .,p** displays the addressed lines. The last line displayed becomes the new current line. You can append **p** to any command other than **E**, **e**, **f**, **Q**, **q**, **r**, **w**, or **!** to check on the effect of that command.
- Q** quits unconditionally, without checking for buffer modifications.
- q** exits **ed**. If you have made changes to the buffer since the last save, **ed** issues a warning. Entering the **q** command again lets you quit, regardless of unsaved changes.
- \$r** [*file*]
reads the contents of *file* into the buffer after the addressed line. If the address is 0, **ed** places the text before the first line in the buffer. If you do not specify *file*, **ed** uses the *remembered* file name; if no *remembered* file name exists, *file* becomes the new *remembered* name. The **r** command displays the number of bytes read from *file* unless you specified the **-s** option. The last line read from the file becomes the new current line. If *file* is replaced by `!`, the rest of the line is considered a shell command line, the output of which is to be read.
- .,s/regexp/new/[flags]**
searches the specified range of lines for strings matching the regular expression *regexp*. Normally the **s** command replaces the first such matching string in each line with the string *new*. The **s** command sets the current line number to the last line on

which a substitution occurred. If **ed** makes no such replacements, **ed** considers it an error.

flags can be zero or more of the following:

- n** replaces the *n*th matching string in the line instead of the first one.
- g** replaces *every* matching string in each line, not just the first one.
- l** displays the new current line in the format of the **l** command.
- n** displays the new current line in the format of the **n** command.
- p** displays the new current line in the format of the **p** command.

You can use any single printable character other than the space or newline instead of / to separate parts of the command provided that you use the same character to delimit all parts of the command. You may omit the trailing delimiter.

You can include a newline in the *new* string by putting a \ immediately in front of the newline. This is a good way to split a line into two lines. If *new* consists only of the % character, **s** uses the *new* string from the previous **s** command. If & appears anywhere in *new*, **ed** replaces it with the text matching the *regexp*. If you want *new* to contain a literal ampersand or percent sign, put a backslash (\) in front of the & or % character.

..,ta copies the addressed lines to the point *after* the line given by the address *a*. The address *a* must not fall in the range of addressed lines. If address *a* is 0, **ed** copies the lines to the beginning of the buffer. This sets the current line to the last line copied.

u rolls back the effect of the last command that changed the buffer. For the purposes of **u**, commands that change the buffer are: **a**, **c**, **d**, **g**, **G**, **i**, **j**, **m**, **r**, **s**, **t**, **v**, **V**, and (of course) **u**. This means that typing **u** repeatedly flips the most recent change back and forth. **ed** treats all changes made by a global command (**G**, **g**, **V**, or **v**) as a single change. As a result, such changes can be easily undone. This command sets the current line number to the value it had immediately before you issued the command being undone.

1,\$V/regexp/

is similar to the **G** command except that this command only gives you the chance to edit lines that do *not* match the given regular expression.

1,\$v/regexp/commands

is similar to the **g** (global) command except that **ed** only applies the given *commands* to lines that do *not* match the given regular expression.

1,\$W [*file*]

is similar to the **w** command except that this command appends data to the given *file* if the file already exists.

1,\$w [*file*]

writes the addressed lines of the buffer to the named *file*. This does not change the current line number. If you do not provide *file*, **ed** uses the *remembered* file name; if there is no *remembered* file name, *file* becomes the *remembered* name. If the output file does not exist, **ed** creates it. **ed** displays the number of characters written unless you had specified the **-s** option.

X prompts you to enter an *encryption key*. All subsequent **e**, **r**, and **w** commands use this key to decrypt/encrypt text read from or written to files. To turn off encryption, issue an **X** command and press RETURN in response to the prompt for an encryption key.

!*command*

runs *command* as if you typed it to your chosen command interpreter. If *command* contains the % character, **ed** replaces it with the current *remembered* file name. If you want a command to contain a literal %, put a backslash (\) in front of the character. As a special case, typing !! re-issues the previous *command*.

\$= displays the line number of the addressed line. This does not change the current line number.

.+1,.,+1 if you supply zero, one, or two addresses without an explicit command, **ed** displays the addressed lines in the mode of the last display command: **p**, **l**, or **n**. This sets the current line number to the last line displayed.

ENVIRONMENT VARIABLES

ed uses the following environment variables:

COLUMNS

contains the terminal width in columns. **ed** folds lines at that point. If it is not set, **ed** uses the appropriate value from the *TERMINFO* database or if that is not available, it uses a default of 80.

HOME

contains the path name of your home directory.

SHELL contains the full path name of the current shell.

TMPDIR

is the path name of the directory being used for temporary files. If it is not set, MPE/iX Shell and Utilities uses /tmp.

FILES

ed uses the following files:

/tmp/e*

This is the *paging file*. It holds a copy of the file being edited. You can change the directory for temporary files using the environment variable *TMPDIR* (see **environ(3)**).

ed.hup

ed writes the current buffer to this file when it receives a hang-up signal.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 A non-usage error occurs
- 2 Usage error.

Messages

Message: ?

Cause: An error occurred while **ed** was in terse mode. This is the default.

Action: For a more verbose description of the error, type **h**. To switch to verbose mode, type **H**.

Message: Addressed line out of range

Cause: You specified an address for a command that referenced a line that does not exist.

Action: Modify the address given to correctly reference the desired lines.

Message: Badly constructed regular expression

Cause: You made an error in the syntax of a regular expression.

Action: Review the documentation on regular expressions (**regexp(3)**) and correct the offending syntax.

Message: Badly formed name

Cause: You specified an improperly formed or missing file name with a command which requires a file name as an argument (for example, **e** or **f**).

Action: Correct or provide the offending file name.

Message: command not allowed inside g, v, G, or V

Cause: You specified a command that cannot be used with the issued global command (**g**, **v**, **G**, or **V**).

Action: Check the *Commands* subsection of this man page for a list of commands that cannot be used with the various global commands

-
- Message:** Destination cannot straddle source in 'm' and 't'
Cause: You specified a range of lines to be moved or copied by **m** or **t** that included the destination address.
Action: Ensure that the specified range of lines for **m** or **t** does not include the destination address.
- Message:** File *filename*: system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** File read error
Cause: An error occurred while reading a file into the page buffer.
Action: Exit editor and restart. Check for system problems. Contact your system manager.
- Message:** Global command too long
Cause: You specified a global instruction (**g** or **v**) that was longer than 256 characters, including newlines.
Action: Specify a global instruction that is less than 256 characters in length.
- Message:** Illegal command redirection
Cause: You attempted to use the **!** command redirection with the **f** command.
Action: Do not use the **!** command redirection with the **f** command.
- Message:** Illegal command suffix
Cause: You specified a command suffix for a command that does not accept suffixes.
Action: Check the *Commands* subsection of this man page for a list of **ed** commands and their syntaxes.
- Message:** Input line too long
Cause: You entered an **ed** command which was too long.
Action: Simplify the command and try again.
- Message:** Incomplete regular expression.
Cause: You issued a **g** or **G** command but did not provide a regular expression as an argument.
Action: Provide a regular expression as an argument to the offending command.
- Message:** insufficient memory
Cause: There were not enough free system resources to perform the specified operation.
Action: Free up more resources.
- Message:** Line(s) too long -- truncated
Cause: One or more lines in the file being edited were too long for **ed** to handle. Those lines were truncated.
Action: To avoid losing data, exit immediately without saving.

-
- Message:** 'm' and 't' require destination address
Cause: You issued an **m** or **t** command but did not provide a destination address.
Action: Provide a destination address with the **m** or **t** command.
- Message:** Mark name must be lower case
Cause: You attempted to use the **k** command to mark an addressed line with a character other than a lowercase letter.
Action: Use **k** to mark the line with a lowercase letter.
- Message:** Missing trailing delimiter after pattern.
Cause: You specified a pattern as part of a **ed** command but did not delimit it.
Action: Provide a trailing delimiter for the pattern.
- Message:** Name too long
Cause: The file name specified on the **ed** command line was too long.
Action: Use a shorter file name.
- Message:** Need space after command
Cause: You did not separate a command from its file name argument with a space.
Action: Re-enter the command with the required space.
- Message:** No match found for regular expression
Cause: The **/** command failed to find any matching lines.
Action: Try a different regular expression.
- Message:** no memory for line number tables
Cause: There were not enough free system resource to allocate initial resources for **ed**.
Action: Free up more system resources and restart program.
- Message:** no memory for pages
Cause: There were not enough free system resources to allocate initial resources for **ed**.
Action: Free up more system resources and restart program.
- Message:** No remembered file name
Cause: You tried to execute a command that uses a remembered file name when there was no remembered file name.
Action: Issue the command again, but specify a file name this time.
- Message:** No remembered regular expression
Cause: You attempted to use **&** to refer to a remembered regular expression when there was no remembered regular expression.
Action: Issue the command again, but specify a regular expression this time.

-
- Message:** no space for line table
Cause: There were not enough free system resources to allocate initial resources for **ed**.
Action: Free up more system resources and restart program.
- Message:** Only one file name is allowed.
Cause: You specified more than one file name on the command line when you invoked **ed**.
Action: Specify only one file name when invoking **ed**.
- Message:** Out of memory for lines
Cause: **ed** was unable to allocate system resources while trying to insert or append lines to the buffer.
Action: Split the file into small pieces.
- Message:** Restricted shell
Cause: You invoked the restricted form of **ed** (**red**), but then tried to use a command that is not allowed in the restricted editor (the **!** command).
Action: See the *DESCRIPTION* section of this man page for a discussion of the differences between **ed** and **red**.
- Message:** Result line of join too long
Cause: You attempt to use the **j** command to join a range lines into one line; however, the resulting line would be too long for **ed** to handle.
Action: Specify a smaller range of lines to be joined.
- Message:** Result of substitution would produce a line too long
Cause: You specified a replacement string in a substitution command that would produce a line that is too long for **ed** to handle.
Action: Specify a shorter replacement string or split the original line into shorter lines before performing the substitution.
- Message:** Temporary file error
Cause: An error occurred when accessing the paging file (see *FILES*).
Action: See your system manager.
- Message:** Undefined mark name
Cause: You attempted to reference a mark name that you have not assigned.
Action: Use the **k** command
- Message:** Unknown command
Cause: You entered a command that does not exist in **ed**.
Action: Check the *Commands* subsection of this man page for a list of valid **ed** commands.

-
- Message:** Unknown option "*-option*".
Cause: You specified an option that is not valid for **ed**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **ed** options.
- Message:** Wrong number of addresses for command
Cause: You specified the wrong number of addresses for the command that you entered.
Action: Check the *Commands* subsection of this man page for a list of **ed** commands and the number of addresses that you can specify with each.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

The addresses < and >, the **-b** and **-x** options, and the **W** and **X** commands are extensions to the POSIX standard.

The **red** command is also an extension to the POSIX standard.

LIMITS

ed allows a limit of 1024 bytes per line and 500,000 lines per file. It does not allow the NUL character. The maximum length of a global command is 256 characters, including newlines.

MPE/iX NOTES

Although the current MPE/iX implementation of the **ed** command can read non-byte stream files, it can only write byte stream files. As a result, if you edit a non-byte stream file with **ed** and save it, that file is now a byte stream file. File characteristics like file code, record size, and so forth are not preserved by this conversion.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

awk(1), **crypt(1)**, **diff(1)**, **ex(1)**, **grep(1)**, **sed(1)**, **vi(1)**, **environ(3)**, **regexp(3)**

NAME

env — display environment, set environment for process

SYNOPSIS

env [-i] [*variable=value ...*] [*command argument ...*]

env [-] [*variable=value ...*] [*command argument ...*]

DESCRIPTION

If you call **env** with no arguments, it displays the environment that it received from its parent (presumably the shell).

Arguments of the form

variable=value

let you add new variables or change the value of existing variables of the environment.

If you specify *command*, **env** calls *command* with the *arguments* that appear on the command line, passing the accumulated environment to this command. The *command* is executed directly as a program found in the search *PATH*, and is not interpreted by a shell.

Options

env accepts the following options:

-i does not use the environment inherited by **env**.

- obsolescent version of -i.

EXAMPLES

Compare the output of the following two examples which illustrate the use of **env**:

```
env foo=bar env
```

```
env -i foo=bar env
```

ENVIRONMENT VARIABLES

PATH contains a list of directories to search when attempting to find *command*.

DIAGNOSTICS

Possible exit status values are:

0 Successful completion.

1 Failure due to any of the following
— insufficient memory
— name is too long.

-
- 2 An invalid command line argument.
- 126 **env** found *command* but was unable to invoke it.
- 127 **env** was unable to find *command*.

Messages

- Message:** too many environment variables
Cause: You specified more than 512 environment variables in a single **env** command.
Action: Do not use more than 512 environment variables in a single **env** command.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **env**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **env** options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. UNIX System V.

printenv on Berkeley UNIX systems has similar functionality.

MPE/iX NOTES

When a program is invoked from the MPE/iX CI, all MPE/iX CI variables will be exported into the environment of the new process.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

sh(1), **environ**(3)

NAME

eval — evaluate arguments in shell

SYNOPSIS

eval [*argument* ...]

DESCRIPTION

The shell evaluates each argument as it would for any command. **eval** then concatenates the resulting strings, separated by spaces, and evaluates and executes this string in the current shell environment.

EXAMPLE

The command:

```
for a in 1 2 3
do
    eval x$a=fred
done
```

sets variables x1, x2 and x3 to fred. Once this has been done,

```
echo $x1 $x2 $x3
```

produces:

```
fred fred fred
```

DIAGNOSTICS

Possible exit status values are:

0 You specified no arguments or the specified arguments were empty strings.

Otherwise, the exit status of **eval** is the exit status of the command that **eval** executes.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh**(1) man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

NOTE

This is a special built-in command of the shell.

eval(1)

MPE/iX Shell and Utilities

eval(1)

SEE ALSO

exec(1), sh(1)

NAME

ex — text editor

SYNOPSIS

ex [**-eRrsvx**] [**-c** *command*] [**-t** *tag*] [**-w** *size*] [*file* ...]

DESCRIPTION

ex is the line-editor mode of the Vi text editor. It supports the following options:

-c *command*

begins editing by executing the specified editor command. You can specify multiple commands by separating them with an or-bar (|). *command* can be any **ex** command except those that enter input mode, such as **insert** or **append**.

-e invokes **ex**. This option is intended for use with Vi.

-r recovers named files after an editor or system crash.

-R sets read-only mode.

-s suppresses all interactive feedback (quiet mode).

-t *tag* edits the file containing the specified *tag* and sets the virtual position in the edit buffer to point of definition for the tag. (see **ctags**(1))

-v invokes Vi.

-w *size* sets option variable `window` equal to *size*.

-x uses encryption.

For further information, see **vi**(1).

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. Most UNIX systems.

-x is an extension to the POSIX standard.

See the *PORTABILITY* section of the **vi**(1) man page.

MPE/iX NOTES

See **vi**(1) for the complete list of limitations affecting the current MPE/iX implementation of **ex**.

SEE ALSO

ed(1), **vi**(1)

NAME

exec — execute a command in place of the current shell

SYNOPSIS

exec [*command_line*]

DESCRIPTION

The argument to **exec** is a *command_line* for another command. **exec** runs this command without creating a new process. Some people picture this as *overlaying* the command on top of the currently executing shell. When the command exits, control returns to the parent of the shell.

Input and output redirections are valid in the *command_line*. You can modify the input and output descriptors of the shell by giving only input and output redirections in the command. For example,

```
exec 2>errors
```

redirects the standard error stream to `errors` in all subsequent commands run by the shell.

If you do not specify a *command_line*, **exec** simply returns a successful exit status.

DIAGNOSTICS

If you specify *command_line*, **exec** does not return to the shell. Instead, the shell exits with the exit status of *command_line* or one of the following exit status values:

- 1-125 A redirection error occurred.
- 126 The command in *command_line* was found but it was not an executable utility.
- 127 The given *command_line* could not be executed because the command could not be found in the current *PATH*.

If you did not specify *command_line*, **exec** returns with an exit value of zero.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh(1)** man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

NOTE

This is a special built-in command of the shell.

exec(1)

MPE/iX Shell and Utilities

exec(1)

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

sh(1)

NAME

exit — exit from the shell

SYNOPSIS

exit [*expression*]

DESCRIPTION

exit terminates the shell.

On a POSIX-compliant system, the value of *expression* should be between 0 and 255. The EXIT trap is raised by the **exit** command, unless **exit** is being invoked inside an EXIT trap.

DIAGNOSTICS

exit returns the value of the arithmetic *expression* to the parent process as the exit status of the shell. If you omit *expression*, it returns the exit status of the last command executed.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh(1)** man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*OPEN Portability Guide 4.0. All UNIX systems.

Allowing an *expression* rather than just a number is an extension found in the KornShell.

NOTE

This is a special built-in command of the shell.

MPE/iX NOTES

If you are using the MPE/iX Shell under the MPE/iX CI, remember that using the **exit** command at the shell's root level does not log you off the system but merely returns you to the MPE/iX CI.

SEE ALSO

return(1), **sh(1)**

NAME

expand — expand tabs to spaces

SYNOPSIS

expand [-t *tablist*] [*file* ...]

expand [-*number*] [-*number,number...*] [*file* ...]

DESCRIPTION

expand reads text input from the *files* specified on the command line, converts tabs into spaces, and writes the result to the standard output. If you do not specify any *files* on the command line, **expand** reads from the standard input.

expand preserves backspace characters. By default, tab stops are set every eight columns. A tab after the last tabstop is replaced by a space.

Options

The first syntax of **expand** accepts the following option:

-t *tablist*

sets tab stops at positions indicated by *tablist*. Numbers in *tablist* must be in ascending order (origin 0) and separated by commas or blanks; however the list must be one argument so you need shell quoting if you are using blanks. The list may consist of a single number, in which case tabs are set every *tablist* positions apart.

The second syntax of **expand** (which the POSIX standard considers obsolete) accepts the following options:

-number

sets tab stops every *number* columns.

-number,number...

sets tab stops at each column *number* (origin 0).

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** Bad tab stop specification
Cause: You specified an illegal character in a tab stop specification, or you did not specify tab stops in ascending order.
Action: Re-enter the command with a valid tab specification.
- Message:** file "*filename*": system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** insufficient memory
Cause: There were not enough free system resources for **expand** to expand all tabs to spaces.
Action: Free up more system resources.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **expand**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **expand** options.
- Message:** write error on standard output
Cause: See **syserror**(3).
Action: See **syserror**(3).

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. 4.2 BSD UNIX and up.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

pr(1), **unexpand**(1)

NAME

export — mark names for export

SYNOPSIS

```
export [name[=value] ...]
export -p
```

DESCRIPTION

export marks each *name* so that the current shell exports it automatically to the environment of all commands executed from that shell. Exported variables are thus available in the environment to all subsequent commands. Several commands (for example, **cd**(1), **date**(1), **vi**(1)) look at environment variables for configuration or option information.

Variable assignments of the form *name=value* assign *value* to *name* as well as marking *name* for export.

Calling **export** without arguments lists, with appropriate quoting, the names and values of all variables in the format:

```
Variable="value"
```

If you re-input this format to another shell, variables are assigned appropriately but not exported. The **-p** option lists variables in a format suitable for re-input to the shell (see the description of the **-p** option).

Options

export accepts the following option:

-p lists variables in the form

```
export name="value"
```

suitable for re-input to the shell.

DIAGNOSTICS

Possible exit status values:

- 0 Successful completion.
- 1 Failure due to invalid command line argument.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh**(1) man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

export is a special built-in command of the Bourne Shell and KornShell on UNIX systems.

NOTE

This is a special built-in command of the shell.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

cd(1), **date(1)**, **set(1)**, **sh(1)**, **typeset(1)**, **vi(1)**

NAME

expr — evaluate expression

SYNOPSIS

expr *expression*

DESCRIPTION

The set of arguments passed to **expr** constitutes an expression to be evaluated. Each command argument is a separate token of the expression. **expr** writes the result of the expression on the standard output. This command is primarily intended for arithmetic and string manipulation on shell variables.

expr recognizes the following operators. Operators listed together have equal precedence; otherwise, they are in increasing order of precedence. **expr** stores expressions as strings and converts them to numbers during the operation. If the context requires a Boolean value, a numeric value of 0 (zero) or a null string ("") is *false*, and any other value is *true*. Numbers have an optional leading sign, followed by **0** for octal, **0x** for hexadecimal, otherwise decimal, followed by the digits of the number. Numbers are manipulated as long integers.

expr1 | *expr2*

results in the value *expr1* if *expr1* is true; otherwise it results in the value of *expr2*.

expr1 & *expr2*

results in the value of *expr1* if both expressions are true; otherwise it results in 0

expr1 <= *expr2*

expr1 < *expr2*

expr1 = *expr2*

expr1 != *expr2*

expr1 >= *expr2*

expr1 > *expr2*

If both *expr1* and *expr2* are numeric, *[CMD expr] compares them as numbers; otherwise it compares them as strings. If the comparison is true, the expression results in 1; otherwise it results in 0.

expr1 + *expr2*

expr1 - *expr2*

performs addition or subtraction on the two expressions. If either expression is not a number, **expr** exits with an error.

expr1 * *expr2*

expr1 / *expr2*

expr1 % *expr2*

performs multiplication, division, or modulus on the two expressions. If either expression is not a number, **expr** exits with an error.

expr1 : *re*

match *expr1 re*

matches the regular expression *re* against *expr1* treated as a string. The regular expression is the same as that accepted by **ed**, except that the match is always anchored, that is, there is an implied leading `^`; therefore **expr** does not consider `^` to be a metacharacter. If the regular expression contains `\(...\)` and it matches at least part of *expr1*, then **expr** results in only that part; if there is no match, **expr** results in 0. If the regular expression doesn't contain this construct, then the result is the number of characters matched. The function `match` performs the same operation as the colon operator.

substr *expr1 expr2 expr3*

results in the substring of *expr1* starting at position *expr2* (origin 1) for the length of *expr3*.

index *expr1 expr2*

searches for any of the characters in *expr2* in *expr1* and results in the offset of any such character (origin 1), or 0 if no such characters are found.

length *expr1*

results in the length of *expr1*.

(*expr*) groups expressions.

EXAMPLES

```
fname=src/fn_abs.c
expr $fname : '.*_\(.*\)\.c'
```

returns `abs`.

```
a=`expr $a + 1`
```

adds one to the value of the shell variable `a`.

DIAGNOSTICS

Possible exit status values are:

- 0 The result of *expression* is true.
- 1 The result of *expression* is false.
- 2 An error occurred.

Messages

- Message:** divide by 0
Cause: You attempted to divide a number by 0.
Action: Do not divide numbers by 0.
- Message:** internal tree error
Cause: You specified an expression that **expr** was unable to evaluate, due to either syntax errors or unusual complexity.
Action: Correct the syntax errors, or simplify the expression (perhaps by breaking it into parts).
- Message:** no space for expression or string
Cause: There were not enough free system resources for **expr** to allocate for a string or expression.
Action: Simplify the expression.
- Message:** non-numeric argument "*string*"
Cause: You specified a string argument with an operator that requires a numeric argument (that is, +, -, *, /, or %).
Action: Replace the string argument with a numeric argument.
- Message:** regular expression error "*regex*"
Cause: An error occurred while processing the regular expression *regex*.
Action: Check the regular expression.
- Message:** strcoll error, cannot malloc space.
Cause: There are not enough free system resources to allocate string space.
Action: Free up more resources.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

In the shell, **let**(1) largely supercedes this command.

match, **substr**, **length**, and **index** are undocumented on all UNIX systems, though they do appear to exist there. They are extensions to the POSIX standard.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

ed(1), let(1), sh(1), test(1), regexp(3)

NAME

false — fail, quietly

SYNOPSIS

false [*argument ...*]

DESCRIPTION

The **false** command simply returns an exit status of 1 (failure). It ignores any arguments given on the command line. This can be useful in shell scripts.

DIAGNOSTICS

false always returns an exit status of 1.

PORTABILITY

POSIX.2./X/OPEN Portability Guide 4.0. All UNIX systems.

NOTE

This command is provided as both an external utility and a built-in shell utility.

SEE ALSO

sh(1)

NAME

fc — display, fix, edit and re-enter previous commands

SYNOPSIS

fc [-r] [-e *editor*] [*first* [*last*]]

fc -l [-nr] [*first* [*last*]]

fc -s [*old=new*] [*specifier*]

DESCRIPTION

fc displays, edits, and re-enters commands which have been input to an interactive shell. **fc** stands for *fix commands*. The environment variable *HISTSIZE* contains the number of commands that are accessible. If *HISTSIZE* is not defined, 128 commands are accessible.

The shell stores these commands in a history file. When the *HISTFILE* environment variable is defined as the name of a writable file, the shell uses this as the history file; otherwise, the history file is *\$HOME/.sh_history*, if *HOME* is defined and the file is writable. If the *HOME* variable is not defined, or the file is not writable, the shell attempts to create a temporary file for the history. If a temporary file cannot be created, the shell does not keep a history file.

Note: A shell shares history (commands) with all shells that have the same history file. A login shell truncates the history file if it is more than *HISTSIZE* lines long.

Normally, the shell does not keep a history of commands executed from a profile file or the *ENV* file. By default, however, it begins recording commands in the history file when it encounters a function definition in either of these set-up files. This means that the *HISTSIZE* and *HISTFILE* variables must be set up appropriately before the first function definition. If you do not want the history file to begin at this time, use

```
set -o nolog
```

For further information, see **sh**(1) and **set**(1).

Any variable assignment or redirection that appears on the **fc** command line affects both the **fc** command itself and the commands that **fc** produces.

The first form of **fc** in the *SYNOPSIS* section puts you into an editor with a range of commands to edit. When you leave the editor, **fc** inputs the edited commands to the shell.

The first and last command in the range are specified with *first* and *last*. There are three ways to specify a command.

- (a) If the command specifier is an unsigned or positive number, **fc** edits the command with that number.

- (b) If the command specifier is a negative number $-n$, **fc** edits the command that came n commands before the current command.
- (c) If the command specifier is a string, **fc** edits the most recent command beginning with that string.

When you use the first form of the *SYNOPSIS* to edit a command, you can omit either *last* or both *first* and *last*. If you omit *last*, **fc** edits the single command specified by *first*. If you omit both, **fc** edits the previous command that you entered to the shell.

Options

fc accepts the following options:

-e *editor*

invokes *editor* to edit the commands. If you do not specify the **-e** option, **fc** assumes that the environment variable *FCEDIT*, if defined, contains the name of the editor for **fc** to use. If *FCEDIT* is not defined, **fc** invokes **ed**(1) to edit the commands.

-l simply displays the command list. This option does not edit or re-enter the commands. If you omit *last* with this option, **fc** displays all commands from the one indicated by *first* through to the previous command entered. If you omit both *first* and *last*, **fc** displays the 16 most recently entered commands.

-n suppresses command numbers when displaying commands.

-r reverses the order of the commands in the command range.

-s re-enters exactly one command without going through an editor. If a command *specifier* is given, **fc** selects the command to re-enter as described earlier; otherwise, **fc** uses the last command entered. To perform a simple substitution on the command before re-entry, use a parameter of the form

old=new

The string *new* replaces the first occurrence of string *old*. **fc** displays the (possibly modified) command before re-entering it.

ENVIRONMENT VARIABLES

fc uses the following environment variables:

FCEDIT

contains the default editor to be used if none is specified with the **-e** option. If this variable is unset or null, **fc** uses **ed** to edit commands.

HISTFILE

contains the path name of the history file.

HISTSIZE

gives the maximum number of previous commands that are accessible.

DIAGNOSTICS

Possible exit status values are:

- 0 If you specified **-1**, this indicates successful completion.
- 1 Failure due to any of the following:
 - missing history file
 - cannot find the desired line in the history file
 - cannot create temporary file
- 2 An invalid command line option or argument.

If **fc** executes one or more commands, the exit status of **fc** is the exit status of the last executed command.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh(1)** man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

This is a command built into the KornShell on UNIX systems, but not the Bourne Shell. On UNIX systems, the KornShell does not truncate the history file at login.

NOTE

This command is built into the shell. **r** is a built-in alias for **fc -s**. **history** is a built-in alias for **fc -1**.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

alias(1), **ed(1)**, **history(1)**, **print(1)**, **r(1)**, **read(1)**, **sh(1)**, **vi(1)**, **shedit(3)**

NAME

file — determine file type

SYNOPSIS

file [-c] [-f *filelist*] [-m *magic*] *file* ...

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

file makes a guess at the type of each *file* argument by inspecting the attributes and (for an ordinary file) reading an initial part of the file. **file** compares each file on the command line to templates found in a system-maintained *magic* file to determine their file type.

file then divides files which do not match a template in the *magic* file into text files and binary data. Then, by reading an initial segment of the text files and making an informed guess based on the contents, **file** further divides text files into various types such as: C programs, assembler programs, files of commands to the shell, text with **nroff** or other embedded formatting commands, and **yacc** or **lex** programs.

file displays the name of each file along with the file type that it believes the file to be.

Options

file accepts the following options:

-c only checks the template file of magic numbers for validity of format. The magic numbers given in the template file describe particular types of files.

-f *filelist*
examines the files listed in the file *filelist*.

-m *magic*
uses the file *magic* rather than the default file of file-type templates.

FILES

file uses the following file:

`/etc/magic`
default template file of *magic* numbers.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** cannot allocate buffer: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** *filename*: cannot open: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** file "*filename ...*": Line too long
Cause: A line in the file containing the list of file names is too long.
Action: Inspect the file containing the list of file names for invalid input.
- Message:** file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** format error in magic file "*filename*", line *num*
Cause: You specified *filename* as a magic file, but it does not conform to the syntax described in the **magic(2)** man page.
Action: Edit *filename* to fit the magic file syntax, or use a valid magic file.
- Message:** line *num*: bad number in magic file
Cause: The specified line of the magic file does not contain a valid number.
Action: Make sure that fields in the specified magic file are separated by tabs.
- Message:** list file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** magic file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** misplaced > in magic file
Cause: You specified a magic file containing a line beginning with a > that did not follow a regular template line.
Action: Either create a template line before the offending line, or remove the offending line.

-
- Message:** misplaced & in magic file
Cause: You specified a magic file containing a line beginning with a & that did not follow a regular template line.
Action: Either create a template line before the offending line, or remove the offending line.
- Message:** Missing file list
Cause: You specified the **-f** option but did not provide a file list as an argument.
Action: Provide the missing file list.
- Message:** Missing magic file
Cause: You specified the **-m** option but did not provide a magic file as an argument.
Action: Provide the missing argument.
- Message:** no space for read buffer
Cause: There were not enough free system resources for **file** to allocate a read buffer.
Action: Free up more system resources.
- Message:** Only one "-f" option allowed
Cause: You specified multiple **-f** options. Only one is allowed.
Action: Specify only one **-f** option.
- Message:** Option *-option* argument missing
Cause: You did not provide an argument for *-option*.
Action: Provide the missing argument.
- Message:** out of space for magic entries
Cause: There were not enough free system resources for **file** to allocate for entries from the magic file.
Action: Free up more system resources.
- Message:** read error on file "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **file**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **file** options.
- Message:** unknown type or cannot open
Cause: You specified a file that does not exist, cannot be opened, or whose file type cannot be determined.
Action: Check that the file exists and that the file permissions allow access.

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

All options are extensions to the POSIX standard.

MPE/iX NOTES

The current MPE/iX implementation of **file** first examines a file's MPE/iX file code and type and if it is a non-byte stream file, **file** reports this information. If it is a byte stream file, **file** proceeds as described in the *DESCRIPTION* section of this man page.

In addition, this release of MPE/iX does not provide the `lstat()` API. As a result, this command cannot return information on the link itself. It attempts to determine when a symbolic link has been referenced, but can only return the information on the target of the link, rather than the link itself.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

magic(2), **environ(3)**

NAME

find — find files within file tree

SYNOPSIS

find *directory ... expression*

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

find walks down the given file hierarchy starting at *directory*, and finds files which match the criteria given by *expression*. Each directory, file, and special file is checked against *expression*. If you use the **-exec**, **-ok**, or **-cpio** primaries, *expression* has the side-effect of invoking a specified command on each file found. A non-existent *expression* or an *expression* with no side-effects automatically uses the **-print** primary to display the name of any file that matches the criteria of *expression*.

find builds *expression* from a set of primaries and operators; juxtaposition of two primaries implies logical AND operator. You can group primaries and operators using parentheses.

Note: Parentheses are shell metacharacters. To use them in *expression*, you must quote them.

You must delimit all primaries, operators, numbers, arguments, and parentheses with white space. Each *number* noted in the primary list is a decimal number, optionally preceded by a plus (+) or minus (-) sign. If a number is given without a sign, **find** tests for equality; a plus sign implies *greater than* or *older than* and a minus sign implies *less than* or *newer than*.

Operators

find accepts the following operators:

- a** Use between primaries for logical AND. This operator can be omitted with the same result since logical AND is assumed when no operator is used between two primaries.
- o** Use between primaries for logical OR.
- !** Precede *expression* with this operator to negate it.

Primaries

find accepts the following primaries:

- atime** *number*
matches if someone accessed the file during the 24-hour period beginning *number* days ago.

-cpio *cpio-file*

writes the file found to the target file *cpio-file* in **cpio** format. This is equivalent to

```
find ... | cpio -o >cpio-file
```

This primary matches if the command succeeds.

-ctime *number*

matches if someone changed the attributes of the file during the 24-hour period beginning *number* days ago.

-depth

processes directories after their contents. If present, this primary always matches.

-exec *command ;*

takes all arguments between **-exec** and the semicolon as a command line, replacing any argument which is exactly `{ }` (that is, the two brace characters) with the current path name. It then executes the resulting command line, treating a return status of zero from this command as a successful match, non-zero as failure. You must delimit the terminal semicolon with white space.

Note: The semicolon is a shell metacharacter. To use it in *expression*, you must quote it.

-follow

follows symbolic links. If present, this primary always matches.

group *name*

matches if the group owner is *name*. If *name* is not a valid group name, it is treated as a group ID.

-inum *number*

matches if the file has inode number *number*.

-level *number*

does not descend below *number* levels.

-links *number*

matches if there are *number* links to the file.

-mtime *number*

matches if someone modified the file during the 24-hour period beginning *number* days ago.

-name *pattern*

compares the current file name to *pattern*. If there is no match, *expression* fails. The pattern uses the same syntax as file name generation (see **sh**(1)). It attempts to match as many trailing path name components as specified in *pattern*.

-ncpio *cpio-file*

writes the file found to the target file *cpio-file* in **cpio -c** format. This is equivalent to

```
find ... | cpio -oc >cpio-file
```

This primary matches if the command succeeds.

-newer *file*

compares the modification date of the found file to that of the *file* given. This matches if someone has modified the found file more recently than *file*.

-nogroup

matches if no group with a name in the group database owns the file.

-none indicates that some action has been taken; thus **find** does not invoke the default **-print** action. If present, this primary always matches.

-nouser

matches if no user with a name in the user database owns the file.

-ok *command* ;

is similar to **-exec**, but before **find** executes the command, it displays the command to confirm that you want to go ahead. **find** only executes the command line if your input matches the expression for **yes** (yes and no expressions are defined in *LC_MESSAGES*). If you type the expression for **no**, the primary does not match. You must delimit the terminal semicolon with white space.

Note: The semicolon is a shell metacharacter. To use it in *expression*, you must quote it.

-perm [-]*mask*

by default, matches if the permissions on the file are identical to the ones given in *mask*. You may specify *mask* in octal or in symbolic mode (see **chmod**(1)). If you use symbolic mode, **find** assumes that you begin with no bits set in *mask*, and the symbolic mode is a recipe for turning the bits you want on and off. A leading minus sign (-) is special. It means that a file matches if at least all the bits in *mask* are set. As a result, with symbolic mode, you cannot use a *mask* value which begins with a minus sign (-).

If you use octal mode, **find** only uses the bottom twelve bits of the *mask*. With an initial minus sign (-), **find** again matches only if at least all the limits in *mask* are set in the file permissions lists.

-print

displays the current file name. This primary always matches.

-prune

stops traversing deeper into the tree at this point. If present, this primary always matches. **-prune** has no effect if **-depth** is also specified.

-size *number*[*c*]

matches if the size of the file is *number* blocks long, where a block is 512 bytes. If you include the suffix *c*, the file size is *number* bytes.

-type *c*

matches if the type of the file is the same as the type given by the character *c*. Possible values of the character are:

- b – block-special
- c – char-special
- d – directory
- f – regular file
- l – symbolic link
- n – network file
- p – FIFO (named pipe)
- s – socket

-user *name*

matches if the owner of the file is *name*. *Name* can also be a user ID number.

-xdev does not cross device boundaries from the root of the tree traversal. If present, this primary always matches.

ENVIRONMENT VARIABLES

find uses the following environment variable:

PATH

determines the location of the *command* specified with the **-exec** or **-ok** primaries.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** `"-type c" is invalid`
Cause: You specified the `-type` primary but did not follow with a valid character to represent the file type.
Action: Check the *DESCRIPTION* section for a list of valid characters for use with the `-type` primary.
- Message:** `bad number specification in "string"`
Cause: You specified an option that takes a numeric value (for example, `-atime`, `-ctime`), but you did not specify a valid number after the option.
Action: Ensure that options that take a numeric value are followed by a valid number (only decimal digits, preceding by an optional plus or minus sign).
- Message:** `cannot access file "filename": system error`
Cause: See `syserror(3)`.
Action: See `syserror(3)`.
- Message:** `cannot execute "filename": system error`
Cause: See `syserror(3)`.
Action: See `syserror(3)`.
- Message:** `cannot stat file "filename" for -newer: system error`
Cause: See `syserror(3)`.
Action: See `syserror(3)`.
- Message:** `error reading directory "pathname"`
Cause: You attempted to read the directory *pathname*. You do not have read permissions on this directory.
Action: If you need to access the directory *pathname*, see your system manager about acquiring read permissions for that directory. If you do not need to access it, no corrective action is required.
- Message:** `group name "name" is unknown`
Cause: You specified the `-group` primary but did not specify a valid group name.
Action: Specify a valid group name after the `-group` primary.
- Message:** `Insufficient memory`
Cause: There were not enough free system resources to perform the specified operation.
Action: Free up more resources.
- Message:** `must specify a command after -exec/-ok`
Cause: You specified either the `-exec` or the `-ok` primary without specifying a command to be performed.
Action: Provide the missing command.

-
- Message:** must specify option after *-primary*
- Cause:** You specified *-primary*, but did not provide the argument that it requires.
- Action:** Specify a valid argument after *-primary*.
- Message:** non-terminated "*primary*" argument list
- Cause:** You specified the **-exec** or **-ok** primary and did not terminate the argument list following it with a semicolon (;).
- Action:** Terminate the argument list following **-exec** or **-ok** with a semicolon.
- Message:** Octal mode may contain only digits [0-7] in *numstring*
- Cause:** When using the octal mode to indicate new access permissions, you specified a string *numstring* which contained a character other than the digits 0 to 7.
- Action:** Ensure that all octal *mode* values are valid octal numbers.
- Message:** unable to access "*pathname*": *system error*
- Cause:** See **syserror**(3).
- Action:** See **syserror**(3).
- Message:** unable to allocate memory for expression tree
- Cause:** **find** requires system resources to build an expression tree. There were not enough free resources to do so.
- Action:** Free up more system resources or specify a less complex *expression*.
- Message:** Unknown option "*-option*"
- Cause:** You specified an option that is not valid for **find**.
- Action:** Check the *DESCRIPTION* section for a list of valid **find** options.
- Message:** Unknown or missing operator in symbolic mode "*modestring*"
- Cause:** When using the symbolic mode to indicate new access permissions, you specified a string *modestring* which had either a missing or unrecognized operator.
- Action:** Make sure that all *mode* values in symbolic mode contain one of the following operators: +, -, or =.
- Message:** user name "*user*" is unknown
- Cause:** You specified the **-user** primary, but did not provide a valid user name.
- Action:** Provide a valid user name after the **-user** primary.
- Message:** write error on standard output: *system error*
- Cause:** See **syserror**(3).
- Action:** See **syserror**(3).

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

Most UNIX systems do not have a default action of **-print**; hence, they do not need the **-none** option. The **-a** operator is undocumented on many UNIX systems. The **-cpio**,

-follow, **-inum**, **-level**, **-ncpio**, and **-none** primaries are extensions to the POSIX standard.

MPE/iX NOTES

This release of MPE/iX does not provide the `lstat()` API. As a result, this command cannot return information on the link itself. It attempts to determine when a symbolic link has been referenced, but can only return the information on the target of the link, rather than the link itself.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

chmod(1), **cpio(1)**, **sh(1)**

NAME

fmt — simple text formatter

SYNOPSIS

fmt [-bCc jns] [-l *n*] [-p *n*] [-w *n*] [*file* ...]

DESCRIPTION

fmt is a simple text formatter intended for modest tasks such as formatting mail messages, and for use within the Vi family of text editors. Formatted output is written to the standard output. **fmt** reads input from the *file* arguments on the command line; when there are none of these, it reads from the standard input.

In the output, white space at the beginning of a line consists of tab characters and blanks. White space in the middle of a line is always spaces. (which means that **fmt** converts tabs to spaces).

Normally, **fmt** ignores line breaks in input, filling output lines. For example,

```
Mary had a little lamb
  Its fleece was white as snow
```

becomes

```
Mary had a little lamb Its fleece was white as snow
```

However, blank lines, changes in indent, and input lines starting with a . do cause a break. In this way, **nroff/troff** files are preserved.

Options

fmt accepts the following options:

- b assumes block paragraphs (that is, paragraphs are uniformly indented including first line), and breaks lines on every change of input indentation.
- C centers the input lines. **fmt** ignores the indentation of input lines.
- c assumes crown paragraphs (that is, paragraphs start with a line with a negative indent (every line but the first is indented)) and therefore breaks on a decrease in indentation only.
- j produces a justified right margin by inserting extra blanks into output lines as necessary.
- l *n* sets the maximum line length to *n* characters. By default, **fmt** produces output lines of at most 72 characters.

-
- n** ignores indent and inter-word space of input lines and squeezes multiple spaces into one. Normally, **fmt** preserves indentation and inter-word spacing of input lines on output.
 - p n** sets the output page offset to *n* characters (default 0). **fmt** adds this offset to the prevailing line indent.
 - s** does not join short lines to form longer lines. This prevents sample lines of code and other such formatted text from being unduly combined.
 - w n** is identical to the **-l n** option.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

Message: file "*filename*": *system error*

Cause: See **syserror**(3).

Action: See **syserror**(3).

Message: length "*number*" is invalid

Cause: You specified a line length that was less than zero or greater than the maximum length that MPE/iX Shell and Utilities supports (as given by the configuration variable `LINE_MAX`).

Action: Use a line length in the range zero to `LINE_MAX`.

Message: Missing line length.

Cause: You specified the **-l** option but did not provide a line length as an argument.

Action: Provide the missing line length.

Message: Missing offset.

Cause: You specified the **-p** option but did not provide an output page offset as an argument.

Action: Provide the missing output page offset.

Message: Unknown option "*-option*"

Cause: You specified an option that is not valid for **fmt**.

Action: Check the *DESCRIPTION* section of this man page for a list of valid **fmt** options.

PORTABILITY

4.2 BSD UNIX and up.

LIMITS

Does not center and justify simultaneously — centering takes priority.

Does not hyphenate.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

mailx(1), **vi(1)**

NAME

fold — break lines into shorter lines

SYNOPSIS

fold [**-bs**] [**-w** *width*] [**-width**] [*file...*]

DESCRIPTION

fold reads the standard input, or each *file*, if you specify any. Each input line is broken into lines no longer than *width* characters. If you do not specify *width* on the command line, the default line length is 80. The output is sent to the standard output.

Options

fold accepts the following options:

- b** specifies the *width* in bytes rather than in column positions; that is, **fold** does not interpret tab, backspace, and carriage return characters.
- s** breaks each line at the last blank within *width* column positions. If there is no blank that meets the requirement, **fold** breaks the line normally.
- w** *width* specifies a maximum line length of *width* characters.
- width** is identical in effect to **-w** *width*.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** file "*filename*" is binary
- Cause:** You specified the binary file *filename* as a **fold** input file. **fold** only works on text files.
- Action:** Only specify text files as **fold** input files.
- Message:** input file "*filename*": *system error*
- Cause:** See **syserror**(3).
- Action:** See **syserror**(3).
- Message:** Missing width after **-w**
- Cause:** You specified the **-w** option without provide the width argument.
- Action:** Provide the missing width.

Message: input file "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: read error on file "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **fold**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **fold** options.

Message: write error on standard output: *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. 4.2 BSD UNIX.

The *-width* option is an extension to the POSIX standard.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

pr(1)

NAME

frombyte — convert a byte stream files to MPE record files

SYNOPSIS

```
frombyte -b [bytestream_file] record_file
```

DESCRIPTION

The **frombyte** utility copies a byte stream file to an MPE record file. It creates either a fixed-record 80-byte ASCII file (MPE text file) or a fixed-record 120-word binary file. If *bytestream_file* is omitted, **frombyte** reads from standard input.

If more flexibility is needed in file conversions, refer to the description of **FCOPY** in the *MPE/iX Reference Supplement* (32650-90353) or the *FCOPY Reference Manual* (32212-90003).

Options

-b creates a binary output file. If this option is not specified, the output is an ASCII file.

EXAMPLES

To convert a byte stream file to an MPE text file, use the following command:

```
frombyte /usr/src/zork.c zork.c.sys
```

The following example uses **tar** to create an archive of all files in the `/usr/src` directory and pipes the archive through **frombyte** to create an MPE binary file named `tarfile`.

```
tar -cvf - /usr/src | frombyte -b tarfile
```

DIAGNOSTICS

- 0 Successful completion.
- 1 An error occurred.

Messages

Message: close of "*filename*" failed
Cause: An error occurred while closing the output file as a new permanent file.
Action: Make sure that you have the necessary resources and permissions to create a new permanent file.

Message: unable to open input file "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

-
- Message:** unable to open output file "*filename*"
Cause: **frombyte** was unable to open the output file. ACTION Make sure that the necessary file system structures exist and that you have appropriate permissions to create a new file, or purge an existing file with the same name if it exists.
- Message:** unknown option "*-option*"
Cause: You specified an option that is not valid for **frombyte**
Action: Check the *DESCRIPTION* section for a list of valid **frombyte** options.
- Message:** write failed on output file "*filename*"
Cause: An error occurred while writing a record to the output file.
Action: Make sure that there are sufficient system resources available to write the file and that none of the file system limits are being exceeded.

PORTABILITY

The **frombyte** utility is unique to MPE/iX Shell and Utilities.

SEE ALSO

tobyte(1)

MPE/iX NOTES

frombyte is available as both a built-in shell utility and an external utility.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

NAME

functions — display or modify shell functions

SYNOPSIS

```
functions [-tux] name name ...
```

DESCRIPTION

functions lets you modify the attributes of the functions specified by the list of *names* in the command line. It is a built-in alias of **sh**(1) defined with

```
alias functions='typeset -f'
```

If no function *names* are specified, **functions** displays all currently defined functions with the attributes specified by the options. If no options are given, **functions** lists all currently defined functions.

Options

functions accepts the following options:

- t turns on the `xtrace` option for the given functions. See **set**(1).
- u allows you to specify attributes for functions which are not yet defined.
- x marks a function for automatic export. See **export**(1).

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to an invalid command line argument.

If you use the command to display the values of variables, the exit status value is the number of names that are invalid.

Messages

Because this command is an alias for a utility built into the MPE/iX Shell, see the **sh**(1) man page for a complete list of error messages that you may receive when using it.

PORTABILITY

x/OPEN Portability Guide 4.0.

On UNIX systems, **functions** is built into the KornShell but not the Bourne Shell.

NOTE

This is an alias built into the shell.

functions(1)

MPE/iX Shell and Utilities

functions(1)

SEE ALSO

export(1), set(1), sh(1), typeset(1)

NAME

getconf — display POSIX configuration information

SYNOPSIS

getconf *parameter_name* [*pathname*]

DESCRIPTION

getconf writes the value of a configuration variable to the standard output. **getconf** displays numeric values in decimal format and non-numeric values as simple strings. If the value is undefined, **getconf** displays it as the string `undefined`.

The following are POSIX.1 standard *parameter_names* that require a *pathname*.

LINK_MAX

maximum number of links that this file can have.

MAX_CANON

maximum number of bytes in the terminal's canonical input queue (before line editing).

MAX_INPUT

space available in terminal input queue.

NAME_MAX

largest file name size.

PATH_MAX

number of bytes in a path name.

PIPE_BUF

largest atomic write to a pipe.

_POSIX_CHOWN_RESTRICTED

restrictions apply to file ownership changes.

_POSIX_NO_TRUNC

if set, it is error for any path name component to be longer than `NAME_MAX` bytes.

_POSIX_VDISABLE

processes are allowed to disable terminal special characters.

The following are POSIX.1 standard names which do not require a *pathname*.

ARG_MAX

maximum length of arguments for running a program, including environment.

CHILD_MAX
maximum number of simultaneous processes allowed per real user.

CLK_TCK
number of intervals per second in machine clock.

NGROUPS_MAX
number of simultaneous group IDs, per process.

OPEN_MAX
number of open files at any time, per process.

STREAM_MAX
number of streams that one process can have open at one time.

TZNAME_MAX
maximum number of bytes supported for the name of a time zone (not of the *TZ* variable).

PATH
standard *PATH* setting.

_CS_PATH
standard *PATH* setting.

_POSIX_ARG_MAX
minimum conforming value for *ARG_MAX*.

_POSIX_CHILD_MAX
minimum conforming value for *CHILD_MAX*.

_POSIX_JOB_CONTROL
POSIX job control supported.

_POSIX_LINK_MAX
minimum conforming value for *LINK_MAX*.

_POSIX_MAX_CANON
minimum conforming value for *MAX_CANON*.

_POSIX_MAX_INPUT
minimum conforming value for *MAX_INPUT*.

_POSIX_NAME_MAX
minimum conforming value for *NAME_MAX*.

`_POSIX_NGROUPS_MAX`
minimum conforming value for `NGROUPS_MAX`.

`_POSIX_OPEN_MAX`
minimum conforming value for `OPEN_MAX`.

`_POSIX_PATH_MAX`
minimum conforming value for `PATH_MAX`.

`_POSIX_PIPE_BUF`
minimum conforming value for `PIPE_BUF`.

`_POSIX_SAVED_IDS`
processes have saved set-user-ID and saved set-group-ID.

`_POSIX_SSIZE_MAX`
value that can be stored in an object of type `ssize_t`.

`_POSIX_STREAM_MAX`
minimum conforming value for `STREAM_MAX`.

`_POSIX_TZNAME_MAX`
minimum conforming value for `TZNAME_MAX`.

`_POSIX_VERSION`
gives version of POSIX adhered to in this release.

The following are POSIX.2 standard names that do not require a *pathname*.

`BC_BASE_MAX`
maximum `ibase` and `obase` values for the **bc** command.

`BC_DIM_MAX`
maximum number of elements permitted in a **bc** array.

`BC_SCALE_MAX`
maximum `scale` size allowed in **bc**.

`BC_STRING_MAX`
maximum number of characters in a string in **bc**.

`COLL_WEIGHTS_MAX`
maximum number of weights assignable to an entry of the `LC_COLLATE` order keyword.

`EXPR_NEST_MAX`
maximum number of expressions that you can nest inside parentheses in an expression evaluated by **expr**.

-
- LINE_MAX**
maximum number of characters that a utility can accept as an input line (either from the standard input or a text file), when the utility takes text files as input. This number includes the trailing newline.
- RE_DUP_MAX**
maximum number of repeated occurrences of a regular expression when using the interval notation $\{m,n\}$ (see **regexp(3)**).
- POSIX2_C_BIND**
indicates if the system supports the C Language Bindings Option.
- POSIX2_C_DEV**
indicates if the system supports the C Language Development Utilities Option.
- POSIX2_FORT_DEV**
indicates if the system supports the FORTRAN Development Utilities Option.
- POSIX2_FORT_RUN**
indicates if the system supports the FORTRAN Runtime Utilities Option.
- POSIX2_LOCALEDEF**
indicates if the system supports the creation of locales.
- POSIX2_SW_DEV**
indicates if the system supports the Software Development Utilities Option.
- POSIX2_CHAR_TERM**
indicates if the system supports at least one terminal type capable of all operations necessary for the User Portability Utilities. Only on if **POSIX2_UPE** is on.
- POSIX2_UPE**
indicates if the system supports the User Portability Utilities Option.
- POSIX2_VERSION**
gives the version of POSIX.2 adhered to in this release.
- POSIX2_BC_BASE_MAX**
minimum conforming value for **BC_BASE_MAX**.
- POSIX2_BC_DIM_MAX**
minimum conforming value for **BC_DIM_MAX**.

POSIX2_BC_SCALE_MAX
minimum conforming value for BC_SCALE_MAX.

POSIX2_BC_STRING_MAX
minimum conforming value for BC_STRING_MAX.

POSIX2_COLL_WEIGHTS_MAX
minimum conforming value for EQUIV_CLASS_MAX.

POSIX2_EXPR_NEST_MAX
minimum conforming value for EXPR_NEST_MAX.

POSIX2_LINE_MAX
minimum conforming value for LINE_MAX.

POSIX2_RE_DUP_MAX
minimum conforming value for RE_DUP_MAX.

This implementation of **getconf** also recognizes the following non-POSIX-compliant name.

_CS_SHELL
default shell (command interpreter).

EXAMPLES

The following example uses **getconf** to find the largest scale value supported by the MPE/iX **bc** utility. If you enter

```
getconf BC_SCALE_MAX
```

getconf displays

```
32767
```

DIAGNOSTICS

Possible exit status values are:

- 0 The specified *parameter_name* was valid and **getconf** displayed its value successfully.
- >0 An error occurred.

Messages

Message: *argument: system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

Message: insufficient memory for buffer.

Cause: There were not enough free system resources for **getconf** to allocate to its buffer.

Action: Free up more resources.

Message: Unknown option "*-option*"

Cause: You specified an option that is not valid for this command.

Action: Check the *DESCRIPTIONS* section for a list of valid options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

`_CS_SHELL` is an extension to the POSIX standard. Some symbols are only supported on systems that support POSIX.2, Draft 8. Some symbols only exist on POSIX.1 systems later than 1990.

SEE ALSO

`bc(1)`, `expr(1)`, `sh(1)`, `regexp(3)`

NAME

getopt — external command to parse shell file options

SYNOPSIS

```
getopt [-c cmdname] optiondesc argument ...
```

DESCRIPTION

The **getopt** command is often used in shell scripts to parse command line options. The first command argument, *optiondesc*, contains each option letter that is valid in the following command *argument* strings. An option letter followed by a colon (:) means that the preceding option letter requires a further *argument* (as in `-o file`).

getopt considers each *argument* that begins with a `-` a potential option, and prints an error if it does not find the *argument* in *optiondesc*. Scanning for further options stops at the first *argument* which does not begin with `-` or with an argument that is `--`. In either case, the options are separated from the rest of the non-option *argument* strings by a `--` string.

The most common construct for using **getopt** is

```
set -- $(getopt [-c cmdname] optiondesc "$@")
```

This may be used inside the MPE/iX Shell to parse the arguments to a shell script; see **sh**(1) for more about the shell.

Options

getopt accepts the following option:

`-c cmdname`

uses *cmdname* rather than **getopt** when displaying error messages.

EXAMPLE

The command:

```
getopt -c diff befhnD: -eh -D string file1 file2
```

which parses the **diff** command line options, would produce the following output:

```
-e -h -D string -- file1 file2
```

The following is a more realistic and complex example of using **getopt** in a shell script.

```

# Example illustrating use of getopt command. This
# shell script would implement the paste command,
# using getopt to process options, if the underlying
# functionality was embedded in hypothetical utilities
# hpaste and vpaste, which perform horizontal and
# vertical pasting respectively.
#
paste=vpaste      # default is vertical pasting
seplist="  "      # default separator is tab

set -- $(getopt -c $0 d:s "$@")
if [ $? -ne 0 ]
then print >&2 "Usage: $0 [-s] [-d seplist] file ..."
    exit 1
fi
for o
do case "$o" in
    -d) shift; seplist="$1"; shift;;
    -s) paste=hpaste; shift;;
    --) shift; break;;
    esac
done

# perform actual paste command
$paste -d "$seplist" "$@"

```

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** Option *-option* argument missing
Cause: You specified *-option* but did not provide the argument that *optiondesc* indicated.
Action: Provide the missing argument.
- Message:** Missing *-c* cmd
Cause: You specified the *-c* option but did not provide a command name as its argument.
Action: Provide the missing argument.

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **getopt**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **getopt** options.

PORTABILITY

UNIX System V.

SEE ALSO

diff(1), **getopts(1)**, **sh(1)**

NAME

getopts — parse options from shell script command line

SYNOPSIS

getopts *optstring name* [*arg ...*]

DESCRIPTION

getopts obtains options and their arguments from a list of parameters that follows the standard POSIX.2 option syntax (that is, single letters preceded by a **-** and possibly followed by an argument value). Typically, shell scripts use **getopts** to parse arguments passed to them. When you specify *args* on the **getopts** command line, **getopts** parses those arguments instead of the script command line (see **set(1)**).

The *optstring* gives all the option letters that the script recognizes. For example, if the script recognizes **-a**, **-f**, and **-s**, *optstring* is *a fs*. If you want an option letter to be followed by an argument value or group of values, put a colon after the letter, as in *a: fs*. This indicates that **getopts** expects the **-a** option to have the form

-a *value*

Normally one or more blanks separate the *value* from the option letter; however, **getopts** also handles *values* that follow the letter immediately, as in

-a*value*

optstring can not contain the question mark (?) character.

The *name* on the **getopts** command line is the name of a shell variable. Each time you invoke **getopts**, it obtains the next option from the positional parameters and places the option letter in the shell variable *name*.

getopts places a question mark (?) in *name* if it finds an option that does not appear in *optstring*, or if an option *value* is missing.

Each option on the script command line has a numeric *index*. The first option found has an index of 1, the second has an index of 2, and so on. When **getopts** obtains an option from the script command line, it stores the index of the script in the shell variable **OPTIND**.

When an option letter has an associated argument (indicated with a **:** in *optstring*), **getopts** stores the argument as a string in the shell variable **OPTARG**. If an option doesn't take an argument or **getopts** expects an argument but doesn't find one, **getopts** unsets **OPTARG**.

When **getopts** reaches the end of the options, it exits with a status value of 1. It also sets *name* to the character ? and sets OPTIND to the index of the first argument after the options. **getopts** recognizes the end of the options by any of the following conditions:

- an argument that doesn't start with -
- the special argument --, marking the end of options
- an error (for example, an unrecognized option letter)

OPTIND and OPTARG are local to the shell script. If you want to export them, you must do so explicitly. If the script invoking **getopts** sets OPTIND to 1, it can call **getopts** again with a new set of parameters, either the current positional parameters or new *arg* values.

By default, **getopts** issues an error message if it finds an unrecognized option or some other error. If you do not want such messages printed, specify a colon as the first character in *opstring*.

EXAMPLE

This is an example of using **getopts** in a shell script. Compare it to the **getopt** example.

```
# Example illustrating use of getopts builtin. This
# shell script would implement the paste command,
# using getopts to process options, if the underlying
# functionality was embedded in hypothetical utilities
# hpaste and vpaste, which perform horizontal and
# vertical pasting respectively.
#
paste=vpaste      # default is vertical pasting
seplist="  "      # default separator is tab

while getopts d:s o
do   case "$o" in
      d)   seplist="$OPTARG";;
      s)   paste=hpaste;;
      [?]) print >&2 "Usage: $0 [-s] [-d seplist] file ..."
            exit 1;;
      esac
done
shift $OPTIND-1

# perform actual paste command
$paste -d "$seplist" "$@"
```

ENVIRONMENT VARIABLES

getopts uses the following environment variables:

OPTARG stores the value of the option argument found by **getopts**

OPTIND contains the index of the next argument to be processed.

DIAGNOSTICS

Possible exit status values are:

- 0 **getopts** found a script command line with the form of an option. This happens whether or not it recognizes the option.
- 1 **getopts** reached the end of the options, or an error occurred.

Message

Because this utility is built into the MPE/iX Shell, see the **sh(1)** man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

On UNIX systems, **getopts** is built into both the KornShell and Bourne Shell.

NOTE

This command is built into the shell.

MPE/iX NOTES

The current release of MPE/iX uses the INFO string to pass arguments to programs. If the size of this string plus the size of the current environment (determined by the number and size of the environment variables in the current process) is greater than 8192 bytes, the string is too long to pass to a subprocess and the process creation fails.

SEE ALSO

sh(1), **getopt(1)**

NAME

grep, egrep, fgrep — match patterns in a file

SYNOPSIS

```
egrep [-bcilnqsvx] [-e pattern] ... [-f patternfile] ... [pattern] [file ...]  
fgrep [-bcilnqsvx] [-e pattern] ... [-f patternfile] ... [pattern] [file ...]  
grep [-bcEFilnqsvx] [-e pattern] ... [-f patternfile] ... [pattern] [file ...]
```

DESCRIPTION

fgrep searches files for one or more *pattern* arguments. It does not use regular expressions; instead, it does direct string comparison to find matching lines of text in the input.

egrep works in a similar way, but uses *extended* regular expression matching, as described in **regex(3)**. If you include special characters in patterns typed on the command line, escape them by enclosing them in apostrophes to prevent inadvertent misinterpretation by the shell or command interpreter. To match a character that is special to **egrep**, put a backslash (\) in front of the character. It is usually simpler to use **fgrep** when you don't need special pattern matching.

grep is a combination of **fgrep** and **egrep**. If you do not specify either **-E** or **-F**, **grep** behaves like **egrep**, but matches *basic* regular expressions instead of extended ones. You can specify a pattern to search for with either the **-e** or **-f** option. If you specify neither option, **grep** (or **egrep** or **fgrep**) takes the first non-option argument as the pattern for which to search. If **grep** finds a line that matches a *pattern*, it displays the entire line. If you specify multiple input files, the name of the current file precedes each output line.

Options

grep accepts all of the following options while **egrep** and **fgrep** accept all but the **-E** and **-F** options.

- b** precedes each matched line with its file block number.
- c** displays only a count of the number of matched lines and not the lines themselves.
- E** causes **grep** to behave like **egrep**.
- e *pattern***
specifies one or more *patterns* separated by newlines for which **grep** is to search.
- F** causes **grep** to behave like **fgrep**.
- f *patternfile***
reads one or more *patterns* from *patternfile*. *Patterns* in *patternfile* are separated by newlines.

-
- i ignores the case of the strings being matched.
 - l lists only the file names that contain the matching lines.
 - n precedes each matched line with its file line number.
 - q suppresses output and simply returns appropriate return code.
 - s suppresses the display of any error messages for nonexistent or unreadable files.
 - v complements the sense of the match; that is, displays all lines *not* matching a pattern.
 - x requires a string to match an entire line.

EXAMPLES

To display every line mentioning an astrological element:

```
egrep "earth|air|fire|water" astro.log
```

DIAGNOSTICS

Possible exit status values are:

- 0 The command found at least one match for *pattern*.
- 1 The command found no matches for *pattern*.
- 2 An error occurred.

If the program fails to open one input file, it tries to go on to look at any remaining input files, but it returns 2 even if it succeeds in finding matches in other input files.

Messages

Message: input file "*filename*": system error

Cause: See **syserror**(3).

Action: See **syserror**(3).

Message: Missing -e pattern

Cause: You specified the **-e** option but did not provide a *pattern* with it.

Action: Provide a *pattern* following the **-e** option.

Message: Missing -f file

Cause: You specified the **-f** option but did not provide a *file* with it.

Action: Provide a *file* following the **-f**.

Message: no room for buffers

Cause: There were not enough free system resources for **grep** to allocate the buffers that it requires.

Action: Free up more system resources.

-
- Message:** out of space for pattern "*string*"
- Cause:** **grep** did not have enough system resources available to store the code needed to work with the given pattern (regular expression). The usual cause is that the pattern is very complex.
- Action:** Make the pattern simpler, or free up more system resources.
- Message:** read error on file "*filename*": *system error*
- Cause:** See **syserror**(3).
- Action:** See **syserror**(3).
- Message:** regular expression error: *regular expression error*
- Cause:** See **regerror**(3).
- Action:** See **regerror**(3).
- Message:** string file "*filename*": *system error*
- Cause:** See **syserror**(3).
- Action:** See **syserror**(3).
- Message:** Unknown option "*-option*"
- Cause:** You specified an option that is not valid for **grep**.
- Action:** Check the *DESCRIPTION* section for a list of valid **grep** options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

Only the actual **grep** command is a part of the POSIX standard. The **egrep** and **fgrep** commands are extensions. The **-b** option is also an extension to the POSIX standard.

LIMITS

The longest input record (line) is restricted by the system variable `LINE_MAX`. It is always at least 2048 bytes. Longer lines are treated as two or more records.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

ed(1), **find**(1), **gres**(1), **regexp**(3)

NAME

hash — create a tracked alias

SYNOPSIS

hash [*name* ...]

DESCRIPTION

hash creates one or more *tracked* aliases. Each *name* on the command line becomes an alias that is resolved to its full path name; thus the shell avoids searching the *PATH* directories for the command whenever you invoke it. A tracked alias is assigned its full path name the first time that the alias is used. It is re-assigned a path name the first time that it is used after the variable *PATH* is changed or the shell command **cd** is used.

hash is a built-in alias defined with

```
alias hash='alias -t'
```

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure because of an invalid command line option.

Messages

Because this command is an alias for a utility built into the MPE/iX Shell, see the **sh**(1) man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

NOTE

This is an alias built into the shell.

SEE ALSO

alias(1), **sh**(1)

NAME

head — display first part of file

SYNOPSIS

head [-bcklmn *num*] [*file* ...]

head [-*num*] [*file* ...]

DESCRIPTION

By default, **head** displays the first 10 lines of each file given on the command line. If you do not specify *file*, **head** reads the standard input.

Options

head accepts the following options:

-**b** *num* displays the first *num* blocks (a block is 512 bytes) of each file.

-**c** *num* displays the first *num* characters of each file.

-**k** *num* displays the first *num* kilobytes (1024 bytes) of each file.

-**l** *num* displays the first *num* lines of each file.

-**m** *num* displays the first *num* megabytes of each file.

-**n** *num* displays the first *num* lines of each file.

-*num* displays the first *num* lines of each file.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - cannot open an input file
 - read error on the standard input
 - write error on the standard output
- 2 Failure due to any of the following:
 - unknown command line option
 - missing or invalid *num* in a -**n** option

Messages

- Message:** Badly formed line/character count "*num*"
Cause: The value *num*, following a **-b**, **-c**, **-k**, **-l**, **-m**, or **-n** option was not a valid number.
Action: Ensure that *num* is a valid number.
- Message:** *filename*: system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** insufficient memory
Cause: There were not enough free system resources to perform the specified operation.
Action: Free up more resources.
- Message:** Missing number after "*option*" option
Cause: You specified the **-b**, **-c**, **-k**, **-l**, **-m**, or **-n** option without providing the value *num*.
Action: Provide the missing number.
- Message:** read error on file "*filename*": system error
Cause: See **syserror**(3).
Action: See **syserror**(3).
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **head**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **head** options.
- Message:** write error on standard output: system error
Cause: See **syserror**(3).
Action: See **syserror**(3).

PORTABILITY

POSIX.2. *x*OPEN Portability Guide 4.0.

The POSIX.2 standard only includes the **-n num** and **-num** options though it considers the latter obsolete.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

cat(1), **sed**(1), **tail**(1)

NAME

help — display brief command explanations

SYNOPSIS

help [*command* ...]

DESCRIPTION

The **help** command provides information about the MPE/iX Shell and Utilities utility specified by *command*.

help acts as a mnemonic reference tool for command options. It offers more information than the standard usage message displayed by all commands, but considerably less than the man pages. Output for a single command is intentionally brief and usually fits on one screen.

help first looks for a file named `/usr/man/man1/command.1`. If such a file exists, **help** looks for lines of the form

```
.HS  
text...  
.HE
```

anywhere in that file, and displays the *text* as the **help** message.

If **help** cannot find this file, it looks in the help file provided in `/etc/helpfile`.

help uses a *helpindex* file to locate information in the help file quickly. If **help** cannot find a *helpindex* file, it creates one in the same directory that holds the help file. If you change the help file (for example, add new information), **help** checks the modification dates and rebuilds the *helpindex* file if the help file has a later date.

You can assign a list of directories (separated by colons) to the environment variable *HELP*. **help** searches these directories (in addition to `/etc`) when looking for the *helpfile* and *helpindex* files. Such directories can hold help information you create yourself.

ENVIRONMENT VARIABLES

help uses the following environment variables:

HELP

contains a list of additional path names to search when looking for *helpfile* and *helpindex* files.

HELPCMD

contains the path name for an additional command to execute if the MPE/iX Shell and Utilities **help** command fails to find help. This lets users add specialized help commands for their environment.

FILES

help uses the following files:

/usr/man/man1/*.1

unformatted manual entries.

/etc/helpfile

help information obtained from man pages.

/etc/helpindex

an index to /etc/helpfile to speed up the search for help information.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure because the help file could not be found or because it contained no information on the desired command.

Messages

Message: No information on *command*

Cause: **help** was unable to find any help information for the specified *command*.

Action: Check that *command* was spelled properly.

PORTABILITY

Some UNIX systems.

MPE/iX NOTES

The **help** command provided by MPE/iX Shell and Utilities should not be confused with the MPE/iX help subsystem.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

man(1), **environ(3)**

NAME

history — display command history

SYNOPSIS

```
history [-nr] [first [last]]
```

DESCRIPTION

history displays commands that you executed previously. These commands make up your *command history*.

By default, **history** displays a numbered list of the 16 most recent commands, from earliest to most recent. By specifying values for *first* and *last*, you can display a specified range of commands rather than the 16 most recent. For example,

```
history 1 10
```

displays commands 1 through 10.

The shell stores your command history in the file given by the variable *HISTFILE*; by default, *.sh_history*. The variable *HISTSIZE* gives the number of commands kept in the file; if *HISTSIZE* is not defined, the default is 128.

Options

history accepts the following options:

- n** displays the commands but not the command numbers.
- r** displays commands in reverse order, from most recent to earliest.

history is an alias defined with

```
alias history='fc -l'
```

For further information, see **fc**(1).

ENVIRONMENT VARIABLES

history uses the following environment variables:

HISTFILE

contains the path name of the history file.

HISTSIZE

gives the maximum number of previous commands that are accessible.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - missing history file
 - could not find the desired command in the history file
- 2 An invalid command line option or argument.

Messages

See the **fc(1)** man page for a list of error messages that **history** may produce.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

NOTE

This is an alias built into the shell.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

alias(1), **fc(1)**, **sh(1)**

NAME

`id` — display user and group names

SYNOPSIS

```
id [user]  
id -G [-n] [user]  
id -g [-nr] [user]  
id -u [-nr] [user]
```

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

Invoking `id` without arguments displays the user name and group affiliations of the person who issues the command. Specifying a *user* argument on the command line displays the same information for the given user instead of the person invoking `id`. In this case, you require appropriate permissions.

The output has the format

```
uid=runum(username) gid=rgnum(groupname)
```

where *runum* is the user's real user ID number, *username* is the user's real user name, *rgnum* is the user's real group ID number, and *groupname* is the user's real group name.

On POSIX-compliant systems, a user's real and effective IDs may differ. In this case, there may be separate entries for effective user ID with the format

```
euid=eunum(euname)
```

where *eunum* is the effective user ID number and *euname* is the effective user name. An entry for effective group ID has the format

```
egid=egnum(egname)
```

where *egnum* is the effective group ID number and *egname* is the effective group name.

Options

`id` recognizes the following options:

- `-G` displays all different group ID's (effective, real, and supplementary) as numbers separated by spaces.
- `-g` displays only the effective group ID number.
- `-n` with `-G`, `-g`, or `-u`, displays the name rather than number.

-
- r** with **-g** or **-u**, displays the real ID rather than the effective one.
 - u** displays only the effective user ID number.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 You specified an invalid user with the **-u** option.
- 2 Failure due to an invalid command line argument.

Messages

- Message:** getgroups failed
- Cause:** See **syserror(3)**.
- Action:** See **syserror(3)**.

- Message:** Insufficient memory
- Cause:** There were not enough free system resources to perform the specified operation.
- Action:** Free up more resources.

- Message:** invalid user name: "*user*"
- Cause:** You specified a user name that was not found in the `passwd` file.
- Action:** Check that you spelled the user name correctly.

- Message:** Unknown option "*-option*"
- Cause:** You specified an option that is not valid for **id**.
- Action:** Check the *DESCRIPTION* section of this man page for a list of valid **id** options.

PORTABILITY

x/OPEN Portability Guide 4.0. UNIX System V.

MPE/iX NOTES

The current release of MPE/iX does not allow a user to belong to more than one group.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

`logname(1)`

NAME

ident — look for keywords in a file

SYNOPSIS

```
ident [-q] [-Ffile...] [file ...]
```

DESCRIPTION

ident searches the named files (or the standard input if no files are specified) for all occurrences of the pattern `$keyword:...$`, where *keyword* is one of the following RCS keywords:

- Author
- Date
- Header
- Id
- Locker
- Log
- Revision
- RCSfile
- Source
- State

The file must be checked out for the **ident** command to work.

The **ident** command works on object files and dumps as well as text files. For example, if the C program in file `f.c` contains

```
char rcsid[] = "$Header:  Header information $";
```

and `f.c` is compiled into `f.o`, the command

```
ident f.c f.o
```

prints

```
f.c:
      $Header:  Header information $
f.o:
      $Header:  Header information $
```

Options

ident accepts the following options:

-Ffile...

provides an alternate way to specify file names. The given *file* is a text file containing a list of file names, one file name per line. **ident** checks all the files named in *file*, using the options specified on the command line. Multiple **-F** options may be

specified on the command line, and can either be grouped together or interspersed between options.

-q suppresses the warning given if there are no keywords in a file.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to an invalid command line argument, or inability to open input file.

Messages

Message: input file "*filename*": *system error*

Cause: See **syserror(3)**.

Action: See **syserror(3)**.

Message: No filename present for **-F** option.

Cause: You specified the **-F** option, but did not provide a file name as its argument.

Action: Provide the missing file name.

Message: Unknown option "*-option*"

Cause: You specified an option that is not valid for **ident**.

Action: Check the *DESCRIPTION* section of this man page for a list of valid **ident** options.

For a list of error messages common to all RCS utilities, see **rcserror(3)**.

PORTABILITY

All UNIX systems.

The **-F** option is an extension to traditional implementations of **ident**.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

ci(1), **co(1)**, **rCS(1)**, **rcsclean(1)**, **rcsdiff(1)**, **rcsmerge(1)**, **rlog(1)**, **rcsfile(3)**

NAME

integer — declare an integer variable

SYNOPSIS

```
integer [±Hlprtux] [±LRZ[number]] [variable[=value] ...]
```

DESCRIPTION

integer declares a shell variable to be an integer. This improves the speed with which the variable can be manipulated. **integer** is a built-in alias defined with

```
alias integer='typeset -i'
```

The options for **integer** are identical to those for the version of **typeset** that deals with variables. Invoking **integer** without any arguments displays all integer variables.

For more information, see **typeset**(1).

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to an invalid command line argument.

If the command is used to display the values of variables, the exit status value is the number of names that are invalid.

Messages

Because this command is an alias for a utility built into the MPE/iX Shell, see the **sh**(1) man page for a complete list of error messages that you may receive when using it.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

On UNIX systems, **integer** is built into the KornShell but not the Bourne Shell.

NOTE

This is an alias built into the shell.

SEE ALSO

alias(1), **sh**(1), **typeset**(1)

NAME

jobs — display status of jobs in current session

SYNOPSIS

jobs [-l|-p] [*job-identifier*...]

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

jobs produces a list of the processes in the current session. Each such process is numbered for easy identification by **fg**(1) and **kill**(1), and is described by a line of information:

[*job-identifier*] *default state shell_command*

job-identifier

is a decimal number which identifies the process for such commands as **fg**(1) and **kill**(1) (preface the *job-identifier* with % when used with these commands).

default identifies the process that is default for the **fg**(1) and **bg**(1) commands (that is, the most recently suspended process). If *default* is a +, this process is the default job. If *default* is a -, this job becomes the default when the current default job exits. There is at most one + job and one - job.

state shows a job as:

Running	if it is not suspended and has not exited
Done	if it exited successfully
Done(<i>exit status</i>)	if it exited with a non-zero exit status
Stopped (<i>signal</i>)	if it is suspended; <i>signal</i> is the signal that suspended the job

shell_command

is the associated shell command which created the process.

Options

jobs accepts the following options:

- l also displays the process group ID of a job (before *state*).
- p displays only the process IDs of all processes.

The -l and -p options are mutually exclusive.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 2 Failure due to an invalid command line argument.

Message

Message: Unknown option "*-option*"

Cause: You specified an option that is not valid for **jobs**.

Action: Check the *DESCRIPTION* section of this man page for a list of valid **jobs** options.

PORTABILITY

POSIX.2. *x/OPEN* Portability Guide 4.0.

MPE/iX NOTES

The current release of MPE/iX does not support job control. As a result, there will never be any jobs in the stopped state. Also, the **fg** and **bg** commands (described in the POSIX.2a standard) are not currently implemented. You can use job IDs with the **kill** command to specify processes.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

kill(1), **wait(1)**

NAME

join — join two sorted, textual relational databases

SYNOPSIS

```
join [-a n] [-e s] [-o list] [-t c] [-v n] [-1 n] [-2 n] file1 file2
join [-a n] [-e s] [-j[n] m] [-o list] [-t c] file1 file2
```

DESCRIPTION

join joins two databases. It assumes that both *file1* and *file2* contain textual databases in which each input line is a record and that the input records are sorted in ascending order on a particular join key field (by default the first field in each file). If you specify **-** in place of *file1* or *file2*, **join** uses the standard input for that file.

Conceptually, **join** computes the Cartesian product of records from both files. By default, spaces or tabs separate input fields and **join** discards any leading or trailing white space. (There can be no white-space delimited empty input fields.) It then generates output for those combined records in which the join key field (the first field by default) matches in each file. The default output for **join** is the common join key field, followed by all the other fields in *file1*, and then all the other fields in *file2*. The other fields from each file appear in the same order they appeared in the original file. The default output field separator is a space character.

Options

join accepts the following options:

- a n** produces an output line for lines that do not match in addition to one for a pair of records that do match. If you specify *n* as one of 1 or 2, **join** produces unpaired records from only that file. If you specify both **-a 1** and **-a 2**, it produces unpaired records from both files.
- e string**
replaces empty fields (selected by **-o**) with *string* on output.
- j[n] m**
uses field number *m* as the join key field. By default, the join key field is the first field in each input line. As with the **-a** option, if *n* is present, this option specifies the key field just for that file; otherwise, it specifies it for both files.
- o list ...**
specifies the fields to be output. You can specify each element in *list* as either *n.m* where *n* is a file number (1 or 2) and *m* is a field number, or as 0 (zero), which represents the join field. You can specify any number of output fields by separating them with blanks or commas. The POSIX-compatible form (listed first in *SYNOPSIS*) requires that you specify multiple output fields as a single argument; therefore shell quoting may be necessary. **join** outputs the fields in the order you list them.

-
- t *c*** sets the field separator to the character *c*. Each instance of *c* introduces a new field, making empty fields possible.
 - v *n*** suppresses matching lines. If you specify *n* as one of 1 or 2, **join** produces unpaired records from only that file. If you specify both **-v 1** and **-v 2**, it produces unpaired records from both files. This does not suppress any lines produced using the **-a** option.
 - 1 *n*** uses the *n*th field of *file1* as the join key field.
 - 2 *n*** uses the *n*th field of *file2* as the join key field.

EXAMPLES

The following script produces a report about files in the current directory containing file name, file mode, and a guess at what the file contains:

```
file * | tr -s ':' ';' >temp1
ls -l | tr -s ' ' ';' >temp2
join -t ';' -2 9 -o 1.1,2.1,1.2 -- temp1 temp2
rm temp[12]
```

This example uses the POSIX implementation of the **join** command. Using the obsolete form of the command, the third line of the script might look like:

```
join -t ';' -j2 9 -o 1.1 2.1 1.2 temp1 temp2
```

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - invalid syntax
 - the wrong number of command line arguments
 - cannot open the input file
 - badly constructed output list
 - too many **-o** options on the command line
- 2 Failure due to an invalid command line argument.

Messages

- Message:** Bad file number specification in "-j"
- Cause:** You specified a file number that was not 1 or 2 with the **-j** option.
- Action:** Specify a file number of 1 or 2 when using the **-j** option.

-
- Message:** Bad join field number
Cause: You specified a value to indicate the join field that was not a valid number.
Action: Make sure to use a valid number to indicate join the field.
- Message:** Badly constructed output list at "*string*"
Cause: You specified an improperly constructed list of output fields with the **-o** option.
Action: Check the *DESCRIPTION* section for details on constructing a list of output fields for the **-o** option.
- Message:** file "*filename*": system error
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Missing **-e** string
Cause: You specified the **-e** option without a string argument.
Action: Provide the missing string.
- Message:** Missing character after **-t**
Cause: You specified the **-t** option without specifying a field separator as an argument.
Action: Provide the missing field separator.
- Message:** Missing join field number
Cause: You specified the **-j**, **-1**, or **-2** option without specifying which field to use as the join field.
Action: Provide the missing join field number.
- Message:** Must specify **-o** with **-e**
Cause: You specified the **-e** option without also specifying the **-o** option.
Action: Always specify the **-o** option when using the **-e** option.
- Message:** Only one character allowed after **-t**
Cause: You specified a field separator that was longer than one character as an argument to the **-t** option.
Action: Use a one character field separator.
- Message:** strcoll error, cannot malloc space.
Cause: There are not enough free system resources to allocate string space.
Action: Free up more resources.
- Message:** too many **-o** list elements
Cause: You specified more than 512 fields in the list of output fields given as the argument to the **-o** option.
Action: Specify less than 512 output fields.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **join**.
Action: Check the *DESCRIPTION* section for a list of valid **join** options.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

POSIX considers the **-j** option to be obsolescent.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

awk(1), **comm(1)**, **cut(1)**, **paste(1)**, **sort(1)**

NAME

kill — terminate process

SYNOPSIS

```
kill -l [exit_status]
kill [-s signal_name] [pid...] [job-identifier...]
kill [-signal_name] [pid...] [job-identifier...]
kill [-signal_number] [pid...] [job-identifier...]
```

DESCRIPTION

kill terminates a process by sending it a signal. The default signal is SIGTERM.

Options

kill accepts the following options:

- l displays the names of all supported signals. If you specify *exit_status*, and it is the exit code of a terminated process, **kill** displays the terminating signal of that process.
- s *signal_name* sends the signal *signal_name* to the process instead of the SIGTERM signal.
- signal_name* is an obsolete equivalent of -s *signal_name*.
- signal_number* is an obsolete method of specifying a positive integer which represents the signal to be used (instead of SIGTERM) as the *sig* argument in the effective call to **kill**. The relationship between *signal_number* and the portable *signal_name* is shown in Table 1-5, *Integer Values of Signals*.

<i>signal_number</i>	<i>signal_name</i>
0	0
1	SIGHUP
2	SIGINT
3	SIGQUIT
6	SIGABRT
9	SIGKILL
14	SIGALRM
15	SIGTERM

Table 1-5: Integer Values of Signals

Operands

kill accepts the following operands:

job-identifier

is the job identifier reported by the shell when a process is started with **&**. It is one way to identify a process. It is also reported by the **jobs** command.

pid

is the process ID that the shell reports when a process is started with **&**. You can also find it using the **ps** command.

The killed process must belong to the current user, unless he or she is the system administrator

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to one of the following:
 - job or process did not exist — error in command line syntax
- 2 Failure due to one of the following:
 - invalid command line argument — invalid signal.

Messages

Message: "*job-identifier*" is not a job

Cause: You specified a job-identifier that is not valid.

Action: Specify a valid job-identifier.

Message: *pid: system error*

Cause: See **syserror(3)**.

Action: See **syserror(3)**.

Message: "*signal*" is not a valid signal

Cause: You specified a non-integer signal for **kill** that was not a valid signal name, or you specified a signal that is outside the range of valid signal numbers.

Action: Make sure that you specify a valid signal number or name for *signal*.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

kill(1)

MPE/iX Shell and Utilities

kill(1)

SEE ALSO

jobs(1), ps(1), sh(1)

NAME

lc — list file system elements in categories

SYNOPSIS

lc [-adfn1] [directory ...]

DESCRIPTION

lc is an alternative to the UNIX **ls** command and the DOS **dir** command. It distinguishes its output by type — generally files or directories. **lc** displays names in columns, sorted horizontally.

Options

lc accepts the following options:

- a display files and directories the names of which begin with .. These path names are not usually shown.
- d displays directories only.
- f displays files only.
- n suppresses all output. This is useful when you only want to use the exit status.
- 1 displays one file per line.

DIAGNOSTICS

Possible exit status values are:

- 0 **lc** successfully found files under the specified *directory* and displayed their names.
- 1 *directory* contains no files or subdirectories. For example, you can use the option combination **-nd** to determine if a directory has any subdirectories.
- 2 Failure because of an invalid command line option or insufficient memory.

Under rare conditions, **lc** cannot identify the type of a file. Such files are listed under the category ****GOK****.

Messages

- Message:** *directory*: Component '*name*' makes path name too long.
Cause: The *name* component of the specified path name makes the overall path name longer than the maximum length indicated by the configuration variable **PATH_MAX**.
Action: Use **cd** to move deeper into the directory structure and try **lc** again. This should result in a shorter overall path name.

-
- Message:** *directory*: Path name component too long
Cause: You specified a *directory* with a name that was longer than the maximum length indicated by the configuration variable `PATH_MAX`. The specified *directory* is invalid, since no directory can exist with a path name that long.
Action: Specify a valid directory.
- Message:** *insufficient memory*
Cause: There were not enough free system resources to perform the requested operation.
Action: Free up more resources.
- Message:** *strcoll error, cannot malloc space.*
Cause: There are not enough free system resources to allocate string space.
Action: Free up more resources.
- Message:** Unknown option "*-option*"
Cause: You specified an option that is not valid for **lc**.
Action: Check the *DESCRIPTION* section of this man page for a list of valid **lc** options.

PORTABILITY

A frequent add-on to UNIX systems.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

c(1), **ls**(1)

NAME

let — evaluate arithmetic expressions

SYNOPSIS

```
let expression ...  
( (expression) )
```

DESCRIPTION

let evaluates each arithmetic *expression* from left to right, using long integer arithmetic with no checks for overflow. No output is generated; the exit status is 0 if the last *expression* has a non-zero value, and 1 otherwise.

The following two lines are equivalent: the second form avoids quoting and enhances readability.

```
let "expression"  
( (expression) )
```

The portable way to write these commands is to the POSIX command:

```
: $( (expression) )
```

Expressions consist of named variables, numeric constants, and operators.

See *Arithmetic Substitutions* in **sh**(1) for syntax of expressions.

EXAMPLE

The commands

```
let a=7 'b=4*2' c=b+1  
echo $a $b $c
```

produce

```
7 8 9
```

DIAGNOSTICS

Possible exit status values are:

- 0 The last argument evaluated to a non-zero value.
- 1 The last argument evaluated to a zero value, or the expression contained a syntax error, or tried to divide by zero.

Messages

Because this utility is built into the MPE/iX Shell, see the **sh**(1) man page for a complete list of error messages that you may receive when using it.

PORTABILITY

`let` is built into the KornShell on UNIX systems, and is not a Bourne Shell command.

NOTE

This command is built into the shell.

SEE ALSO

`sh(1)`, `expr(1)` `test(1)`

NAME

lex — lexical analyzer generator

SYNOPSIS

```
lex [-achlntTv] [-o file.c] [-P proto] [-p prefix] [file.l ...]
```

DESCRIPTION

lex reads a description of a lexical syntax, in the form of regular expressions and actions, from *file.l*. If you do not provide *file.l* or if the file is named `-`, **lex** reads the description from standard input. It produces a set of tables that, together with additional prototype code from `/etc/yylex.c`, constitute a lexical analyzer to scan those expressions. **lex** places this output in the file `lex.yy.c`. The resulting recognizer is suitable for use with **yacc**(1). For detailed information on using **lex**, see the *LEX Programming Guide*.

The LEX library contains a number of functions essential for use with **lex** (described in **lex**(3)). The actual library to use depends on your system and compiler: for example, for UNIX systems and POSIX-compliant systems, use `-ll`. You can specify this as an operand on the **c89** command line (see **c89**(1)). If you use a different compiler, you must create the LEX library yourself from the provided sources.

Some **lex** programs can cause one or more tables within **lex** to overflow. These tables are the NFA, DFA, and move tables; **lex** displays an appropriate message if an overflow occurs. You can change table sizes by inserting the appropriate line into the *definition* section of the lex input, with the number *size* giving the number of entries to use. This is shown in Table 1-6, *Internal Table Sizes*.

Line	Table Size Affected	Default
<code>%esize</code>	number of NFA entries	1000
<code>%nsize</code>	number of DFA entries	500
<code>%psize</code>	number of move entries	2500

Table 1-6: Internal Table Sizes

You can often reduce the NFA and DFA space to make room for more move entries.

Options

lex accepts the following options:

- `-a` lets character classes refer to 8-bit characters (0200 through 0377). Normally, to save table space, character classes only apply to the 7-bit character set. On systems that use 8-bit character sets, this option is on by default, and cannot be turned off.
- `-c` generates C code. As this is the default, this option is only provided for compatibility with other implementations.

-
- h** displays a brief list of the options and quits.
 - l** suppresses `#line` directives in the generated code.
 - n** suppresses the display of table sizes by the **-v** option. If you did not specify **-v** and there are no table sizes specified in *file.l*, **lex** behaves as though you specified **-n**.
 - o file.c**
writes the lexical analyzer (internal state tables) to the named output file, instead of the default `lex.yy.c`.
 - P proto**
uses the named code file, instead of the default prototype file `/etc/yylex.c`.
 - p prefix**
uses the given prefix instead of the prefix `yy` in the generated code.
 - T** writes a description of the analyzer to the file `l.output`.
 - t** writes the lexical analyzer to standard output, instead of the file `lex.yy.c`.
 - v** displays the space used by the various internal tables. Normally, **lex** displays these statistics on the standard output, but if you also specified the **-t** option, it displays them on the standard error. If you did not choose this option and *file.l* specifies table sizes, **lex** still displays these statistics unless you specified the **-n** option.

FILES

`l.output`
scanner machine description

`lex.yy.c`
tables and action routines

`/etc/yylex.c`
the prototype lex scanner

`/usr/lib/libl.a`
UNIX LEX library

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

Messages

- Message:** Badly formed equivalence class [=equiv_class=]
Cause: You attempted to use a multi-character equivalence class in a regular expression. LEX does not support non-POSIX locales.
Action: Rewrite the regular expression.
- Message:** Cannot open temporary file '*filename*'
Cause: LEX was unable to open the temporary file *filename*.
Action: Make sure that *TMPDIR* points to a writable temporary directory, or if *TMPDIR* is not set, that /tmp is a writable temporary directory.
- Message:** Cannot use character class or equivalence class in range
Cause: You attempted to use a character class or an equivalence class (that is, [*:* *:*] or [= =]) in a character range within a regular expression.
Action: Rewrite the regular expression.
- Message:** Collation in [= =] not supported (yet)
Cause: You attempted to use an equivalence class [=collation-symbol.=] within a regular expression. LEX does not (yet) support this construct.
Action: Rewrite the regular expression.
- Message:** Error writing temp file '*filename*'
Cause: An error occurred when LEX tried to write the temporary file *filename*.
Action: Check the directory indicated by *TMPDIR*, or /tmp and ensure that the directory is writable and has sufficient space.
- Message:** Missing output filename after -o
Cause: You specified the -o option without providing the name of an output file.
Action: Provide the missing file name.
- Message:** Missing prefix after -p
Cause: You specified the -p option without providing a prefix as an argument.
Action: Provide the missing prefix.
- Message:** Missing prototype filename after -P
Cause: You specified the -P option without providing a file name as an argument.
Action: Provide the missing file name.

-
- Message:** Multi-character collating element [*.col_element.*] not supported
- Cause:** You specified a regular expression containing a multi-character collating element that is not supported by the POSIX locale.
- Action:** Rewrite the regular expression.
- Message:** No lex rules
- Cause:** You specified LEX input that did not contain any translation rules, possibly due to empty or badly formatted input.
- Action:** Make sure that your input file is specified properly, and that the contents are properly formatted.
- Message:** No more memory for *item*
- Cause:** There were not enough free system resources for LEX to allocate to *item*. Your scanner input was too large or too complicated, or you requested too much space for a table.
- Action:** Simplify your input expressions, or request less space for tables.
- Message:** Out of dfa move space: increase %p from *num*
- Cause:** There were not enough move entries for LEX to process your input.
- Action:** Increase move table size with the %p directive.
- Message:** Out of DFA state space: increase %n from *num*
- Cause:** You did not reserve enough space for the DFA tables.
- Action:** Use the %n directive to increase the space for the DFA tables.
- Message:** Out of NFA state space: increase %e from *num*
- Cause:** You did not reserve enough space for the NFA tables.
- Action:** Use the %e directive to increase the space for the NFA tables.
- Message:** Poorly formed [*char* sequence
- Cause:** You specified a [*. .*], [*= =*], or [*: :*] sequence improperly.
- Action:** Specify the sequence correctly.
- Message:** premature eof in prototype
- Cause:** LEX encountered an end of file character in the prototype file when it was not expecting it, probably due to a badly formatted prototype file.
- Action:** Ensure that the prototype file is not corrupted. If using a private prototype file, ensure that it the same layout as the distributed version.
- Message:** temporary file "*filename*": system error
- Cause:** See **syserror(3)**.
- Action:** See **syserror(3)**.

-
- Message:** Too many character classes (more than *num*)
Cause: LEX ran out of space for character classes.
Action: Simplify your scanner input.
- Message:** Too many move (%p) entries: *num*
Cause: You did not reserve enough space for move tables.
Action: Use the %p directive to increase the space for move tables.
- Message:** Too many translations (more than *num*)
Cause: LEX ran out of space for translation rules.
Action: Simplify your scanner input.
- Message:** Unknown class [:*class*:]
Cause: You specified a regular expression containing a character class [:*class*:] that is not supported in the POSIX locale.
Action: Rewrite the regular expression.
- Message:** Unknown collating element [*.col_element*.]
Cause: You specified a regular expression containing a collating element that is not supported by the POSIX locale.
Action: Rewrite the regular expression.
- Message:** Unknown option: *-option* Try lex -h for help
Cause: You specified an option that is not valid for **lex**.
Action: Check the *DESCRIPTION* section or type **lex -h**, for a list of valid **lex** options.
- Message:** Write error on *filename*
Cause: An error occurred while LEX was writing the output file.
Action: Check that space exists on the output device and that you have appropriate permissions to write the file.

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

The **-a**, **-h**, **-l**, **-o**, **-P**, **-p** and **-T** options are extensions to the POSIX standard.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

yacc(1), **lex**(3)
MPE/iX Shell and Utilities User's Guide

NAME

line — copy one line of standard input

SYNOPSIS

line

DESCRIPTION

line copies one input line from its standard input to its standard output. The end of the line is the first newline encountered. This is useful in shell files that need small amounts of input (for example, responses to prompts).

EXAMPLES

```
echo "Enter name:\c"  
NAME=`line`
```

DIAGNOSTICS

Possible exit status values are:

- 0 A line was read successfully.
- 1 **line** reached end-of-file before finding a newline character.

PORTABILITY

x/OPEN Portability Guide 4.0. UNIX System V.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

cat(1), **head(1)**, **read(1)**, **sh(1)**, **tail(1)**

NAME

ln — create a link to an existing file

SYNOPSIS

```
ln [-fiRrs] old new
ln [-fiRrs] old old ... dir
```

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

ln creates a *link* to an existing file or set of files. A link is a new directory entry that refers to the same file. This entry can be in the same directory that currently contains the file or in a different directory. The result is that you get a new path name that refers to the file. You can access the file under the old path name or the new one. Both path names are of equal importance. If you **rm** either name, the other one still remains and the file contents are still available under that name. The contents of the file do not disappear until you remove the last link.

A file may have any number of links to it. Thus you can establish any number of different path names for any file.

In the first form given in the synopsis, *new* becomes a new path name for the existing file *old*.

In the second form, **ln** creates entries for all the *old* files under the directory *dir*. For example,

```
ln yourdir/* mydir
```

creates links under *mydir* to all the files under *yourdir*. The files have the same names under *mydir* that they had under *yourdir*. **ln** always assumes this directory form when the last operand on the command line is the name of a directory. In this case, none of the *old* names may be a directory.

It is possible that there is already a file that has the same name as the link you are trying to set up. We'll refer to this file as the *conflicting* path name. To deal with a conflicting path name, **ln** follows these steps.

- If you have specified **-i**, **ln** writes a prompt to standard error to ask if you want to get rid of the conflicting path name. If you answer affirmatively, **ln** attempts to remove it.
- Otherwise, if you have specified **-f**, **ln** attempts to remove the conflict silently.
- Otherwise, **ln** prints a diagnostic message.
- **ln** gets to this point if it is going to get rid of the conflicting path name. It therefore attempts to get rid of the conflicting path name in the same way that **rm** does. **ln** deletes the file associated with the path name if this path name is the last link to the file. If **ln**

can't get rid of the conflicting path name, it does not attempt to establish the new link; it writes an error message to the standard error and goes on to the next file.

- If **ln** successfully gets rid of the conflicting path name, it then establishes the link.

Options

ln accepts the following options:

- f** gets rid of any conflicting path names without asking you for confirmation.
- i** checks with you before getting rid of conflicting path names. You must not specify both **-f** and **-i**.
- R** links files recursively. That is, you can link an entire hierarchy of subdirectories at once.
- r** is identical to **-R**.
- s** Creates a symbolic link.

DIAGNOSTICS

Possible exit status values are:

- 0 All requested links were established successfully.
- 1 Failure due to any of the following:
 - an argument had a trailing / but was not the name of a directory
 - a file could not be found
 - an input file could not be opened for reading
 - an output file could not be created or opened for output
 - the new link file already exists
 - a link could not be established
 - a read error occurred on an input file
 - a write error occurred on an output file
 - the input and output files were the same file
 - inability to access a file when using **-r**
 - inability to read a directory when using **-r**
 - inability to create a directory when using **-r**
 - a target is not a directory when using **-r**
 - source and destination directory are the same when using **-r**
- 2 Failure due to any of the following:
 - invalid command line option
 - too few arguments on the command line
 - a target that should be a directory but isn't

- no space left on target device
- out of memory to hold the data to be copied
- the inability to create a directory to hold a target file

Messages

- Message:** cannot allocate I/O buffer: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** cannot allocate target string
Cause: There are not enough free system resources to hold the name of the target file.
Action: Free up more system resources.
- Message:** cannot mkdir "*pathname*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** cannot open file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Cannot reset permissions on file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Cannot reset times on file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Cannot reset uid or gid on file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** Error copying file *file1* to *file2*: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** fifo "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** link to target "*filename*" failed: *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.

-
- Message:** no space on device for file "*filename*"
Cause: You attempted to copy (or move) a file to *filename* on a device that has no space for it.
Action: Free up space on the target device or copy (or move) the file to another device.
- Message:** "*pathname*" is a directory (not copied): *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** read error on file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** recursive copy to directory "*pathname*"
Cause: You tried to recursively link a directory to itself.
Action: Choose a different *pathname*.
- Message:** source "*name1*" and target "*name2*" are identical
Cause: You specified source and target files that are actually the same file (for example, because of links).
Action: No further action is required.
- Message:** special file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** target file "*filename*": *system error*
Cause: See **syserror(3)**.
Action: See **syserror(3)**.
- Message:** target "*pathname*" is not a directory
Cause: When recursively copying (or moving) multiple files using the **-r** or **-R** option, the target must be a directory. You specified a target *pathname* that is not a directory.
Action: Check spelling of target *pathname*.
- Message:** target "*pathname*" must be a directory
Cause: You attempted to copy (or move) two or more files but the target indicated by *name* was not a directory.
Action: When copying (or moving) two or more files, ensure that the final *name* on the command line is a directory.
- Message:** target "*pathname*" must exist
Cause: The destination directory must exist for this utility to work.
Action: Check the command line arguments. You may need to create the target directory.

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for this command.
Action: Check the *Options* section of the command's man page for a list of valid options.

Message: unreadable directory "*pathname*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: write error on file "*filename*": *system error*
Cause: See **syserror**(3).
Action: See **syserror**(3).

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

The **-s** option only works on file systems which support symbolic links.

Only the **-f** option is part of the POSIX standard.

MPE/iX NOTES

This release of MPE/iX does not support the creation of additional hard links to a file. It is only possible to create symbolic links at this time.

In addition, because this release of MPE/iX does not provide the `lstat()` API, this command cannot return information on the link itself. It attempts to determine when a symbolic link has been referenced, but can only return the information on the target of the link, rather than the link itself.

On MPE/iX, **ln** is available as both a built-in shell utility and an external utility.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

cp(1), **locale**(1), **mv**(1), **rm**(1)

NAME

logname — display user name

SYNOPSIS

logname

DESCRIPTION

logname displays the login name of the person who issues the command. It obtains this through the `getlogin()` function defined in the POSIX.1 standard.

ENVIRONMENT VARIABLES

LOGNAME contains your user name.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 **logname** could not determine the login name.

PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. UNIX System V.

MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

env(1), **id(1)**, **environ(3)**

NAME

ls — list file and directory names and attributes

SYNOPSIS

ls [-AabCdFfgiLlMnOpqRrstux1] [pathname ...]

DESCRIPTION

Note: The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

ls lists files and directories. If the *pathname* is a file, **ls** displays information on the file according to the requested options. If it is a directory, **ls** displays information on the files and subdirectories therein. You may obtain information on a directory itself using the **-d** option.

If you do not specify any options, **ls** displays only the file name(s). When **ls** sends output to a pipe or a file, it writes one name per line; when it sends output to the terminal, it uses the **-C** (multi-column) format.

Options

ls accepts the following options:

- A** lists all entries including those starting with periods (.); but excluding any . or .. entries.
- a** lists all entries including those starting with a period (.).
- b** displays non-printable characters in octal, as \ooo.
- C** puts output into columns, sorted vertically; this is the default output format to the terminal.
- c** uses the time of the last modification of the file's attributes for sorting (**-t**) or displaying (**-l**).
- d** does not display the contents of named directories, but information on the directories themselves.
- F** puts a / after each directory name, a * after every executable file, a | after every FIFO file, a @ after every symbolic link, and a = after every socket.
- f** enables the **-a** option and disables the **-C** option. This means that for each argument that is a directory, all directory entries are listed in the same order they are retrieved from the system.
- g** displays only the group ID numbers.

-
- i** displays inode numbers along with file names.
 - L** follows symbolic links (only on systems that support symbolic links).
 - l** displays permissions, links, owner, group, size, time, name; see *Long Output Format*.
 - m** displays names in single line, with commas separating names.
 - n** displays user ID and group ID numbers.
 - o** displays only the user ID of owner.
 - p** puts / after directory names.
 - q** displays non-printable characters as ?.
 - R** lists subdirectories recursively.
 - r** sorts in reverse of usual order; you can combine this with other options that sort the list.
 - s** displays size in blocks (after the inode number, but before other information).
 - t** sorts by time. By default, this option sorts the output by the modification times of files. You can change this with the **-c** and **-u** options.
 - u** uses the last access time for sorting (**-t**) or displaying (**-l**).
 - x** puts output into sorted columns, with output going across the rows.
 - 1** forces output to be single column.

Note: When you specify options that are mutually exclusive (for example, **-c** and **-u**), the option that appears last on the command line is used.

Long Output Format

The output from **ls -l** summarizes all the most important information about the file on one line. If the specified *pathname* is a directory, **ls** displays information on every file in that directory (one file per line). It precedes this list with a status line that indicates the total number of file system blocks (512 byte units) occupied by the files in that directory. Here is a sample of the output along with an explanation.

```
-rw-rw-rw- 1 root  dir 104 Dec 25 19:32 file
```

The first character identifies the file type:

- regular file
- b block special file
- c character special file

d	directory
l	symbolic link
p	FIFO
s	socket

The next nine characters are in three groups of three; they describe the permissions on the file. The first group of three describes owner permissions; the second describes group permissions; the third describes other (or *world*) permissions. Characters that may appear are:

r	Permission to read file
w	Permission to write on file
x	Permission to execute file

The following characters only appear in the execute permission (x) position of the output.

S	Same as s except execute is turned off
s	If in owner permissions section, setuid bit is on; if in group permissions section, setgid bit is on
T	Same as t except execute bit is turned off
t	Save text bit is on

You can set some permissions with the **chmod**(1) command.

After the permissions comes the number of links to the file.

Next comes the name of the owner of the file or directory.

Then comes the name of the group that owns the file or directory.

Following this is the size of the file, expressed in bytes.

After this comes a date and time. For a file, this is the time that the file was last changed; for a directory, it is the time that the directory was created. The **-c** and **-u** options can change which time value is used.

The last item on the line is the name of the file or directory.

ENVIRONMENT VARIABLES

ls uses the following environment variables:

COLUMNS

contains the terminal width in columns. **ls** uses this value to determine the number of output columns to write using the **-C** option.

TZ

contains the time zone to be used when displaying date and time strings.

DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
 - out of memory
 - inability to find a file's information
 - too many directories
 - file/directory not found
- 2 Failure due to an invalid command line option.

Messages

Message: cannot allocate memory for sorting
Cause: There were not enough system resources available for **ls** to sort its output.
Action: Free up more system resources or use option and path names on the command that will produce less output.

Message: File or directory "*pathname*" is not found
Cause: You specified a *pathname* that does not exist.
Action: Check to make sure that you did not omit or misspell any components of *pathname*.

Message: insufficient memory
Cause: There were not enough free system resources to perform the specified operation.
Action: Free up more resources.

Message: strcoll error, cannot malloc space.
Cause: There are not enough free system resources to allocate string space.
Action: Free up more resources.

Message: *pathname* : system error
Cause: See **syserror**(3).
Action: See **syserror**(3).

Message: too many directory entries in "*pathname*"
Cause: This message only appears when **ls** runs out of dynamically allocated system resources.
Action: Free up more system resources.

Message: Unknown option "*-option*"
Cause: You specified an option that is not valid for **ls**.
Action: Check the *DESCRIPTION* section for a list of valid **ls** options.

PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

The **-A**, **-b**, **-f**, **-g**, **-L**, **-m**, **-n**, **-o**, **-p**, **-s**, and **-x** options are extensions to the POSIX standard.

MPE/iX NOTES

On the current MPE/iX implementation, the inode number is MPE/iX's unique 32-bit mapping of the file's 48-bit UFID. Also, on MPE/iX, the user and group fields are 17 characters long and 8 characters long, respectively.

You may notice that for certain files, the current MPE/iX implementation of **ls** shows the date that the file's attributes were last modified to be January 1, 1970. The reason for this is that MPE/iX Release 4.5 added a new field, `state_chg_time` to file labels. This field indicates the time that the file's attributes were modified. When a system is updated to MPE/iX Release 4.5 or later, all files labels are updated to the new data structure; however for files created using MPE/iX releases before 4.5, the system cannot determine what data to put in the `state_chg_time` field and instead, sets the field to zero. **ls** interprets the value zero as January 1, 1970, which is the zero date for all HP 3000 systems. As soon as the file's attributes are modified after the update, the `state_chg_time` field is set to the correct date.

On MPE/iX, **ls** is available as both a built-in shell utility and an external utility.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

SEE ALSO

chmod(1), **lc(1)**

NAME

mailx — read electronic mail

SYNOPSIS

mailx [-efHiNn] [-u *user*] [*filename*]

mailx [-FinU] [-h *number*] [-r *address*] [-s *subject*] *user* ...

DESCRIPTION

mailx helps you read and send electronic mail messages. It has no built-in facilities for sending messages to other systems, but combined with other programs (a mail routing agent, and a transport agent), it can send messages to other systems.

The command line

```
mailx [options] user user user ...
```

sends a mail message to the given users. If you do not specify any users on the command line, **mailx** lets you read incoming mail interactively.

Options

mailx accepts the following options when you are reading messages:

- e checks to see if you have any messages waiting to be read. With this option, nothing is displayed. If you have waiting messages, **mailx** exits with a successful status return; otherwise, **mailx** exits with a failure return.
- f looks for messages in the file given by the optional *filename* on the command line instead of in your system mailbox. If you do not specify *filename*, **mailx** reads messages from ~/mbx.
- H displays only the header summary of a message.
- N does not display the header summary of messages.
- u *user* looks for messages in the system mailbox of the specified user. This only works if you have read permission on the user's system mailbox.

mailx accepts the following options only when you are sending messages:

- F records your message in a file with the same name as the first user specified on the command line. This option overrides the *record* variable, if it has been set. See the *ENVIRONMENT VARIABLES* section for more on the *record* variable.
- h *number* indicates how many *hops* a message has already made from one machine to another (in a network of machines). This option is not intended for most users; some network

mail software uses the option to prevent *infinite loops* (the same message cycling through a sequence of machines without ever getting to its intended destination).

-r *address*

passes the given address to network mail software. If this option is present, it disables all input mode commands.

-s *subject*

uses the given *subject* string in the Subject heading line of the message. If the subject contains spaces or TAB characters, the string should be enclosed in double quotes (" ") or apostrophes (' '). If you specify this option on the command line, **mailx** does not prompt you to enter a Subject line when you type in the text of the message.

-U converts the address from UUCP style to internet standards. This overrides the effect of the *conv* environment variable. See the *ENVIRONMENT VARIABLES* section for more information.

mailx accepts the following options when you are sending or reading messages:

-i ignores interrupts (for example, pressing BREAK or CTRL-C). Also see the description of the *ignore* environment variable in the section on *ENVIRONMENT VARIABLES*.

-n does not initialize your **mailx** session from the system's */etc/mailx.rc* file. For more information about this file, see the *Start-Up Files* section.

General Overview

This section describes the *default* behavior of **mailx**. You can use **mailx** in many ways, and these are discussed later; however, you must first understand **mailx's** *normal* pattern of behavior.

The simplest command to send a message is

```
mailx address address address ...
```

where each *address* names someone who is to receive the message. The simplest kind of address is the *login name* of someone else who uses your machine. You can also use network addresses here, but **mailx** itself cannot send messages over a network; it requires some sort of network server. MPE/iX Shell and Utilities does not include server software.

You can also send messages as input to commands. To do this, use an address that consists of an or-bar (|) followed by a command line that invokes the appropriate command. Enclose this whole address in apostrophes. For example,

```
mail robin '|cat >save'
```

mails a message to *robin* and also copies the message into a file called *save*.

After you type in the command to send a message, **mailx** asks you to enter the Subject of the message (a brief description of what the message is about). You can now type in the text of your message. Your message can consist of any number of lines, and may include blank lines. When you finish entering the message, type a line consisting only of a tilde (~), followed by a dot (.); then press ENTER. This tells **mailx** that the message is ready to be sent.

mailx puts the completed message into a file called the recipient's *system mailbox*. The message stays there until the recipient asks. At that point, the message is obtained from the system mailbox and displayed on the recipient's terminal. It is then saved in the recipient's *personal mailbox*. Since this is usually a file named *mbox* under the recipient's home directory, *mbox* represents the personal mailbox and *mailbox* represents the system mailbox.

The simplest way to read incoming messages is to type the command **mailx** (with no addresses on the command line). This starts an *interactive* session in which **mailx** lets you read your mail and perform other operations. For example, you can display new messages, delete old ones, reply to messages or forward them to someone else, and so on. When you are performing operations in this way, you are in *command mode*. When you are typing in the text of a message, you are in *input mode*.

A message consists of a sequence of *header lines* followed by the body of the message. The header lines tell who sent the message, the time and date that it was sent, its subject, and so on. **mailx** automatically creates header lines. Some of the common header lines are:

Cc: *name name ...*

Stands for *carbon copies*. This indicates that copies of this message are to be sent to the specified recipients. The names of these recipients appear in the header lines of everyone receiving the message.

Bcc: *name name ...*

Stands for *blind carbon copies*. This is similar to **Cc:** but the names of people receiving carbon copies do not appear in the header lines of the message. Recipients do not know that these people received a copy of the message.

Subject: *text*

Gives the subject of the message.

To: *name name ...*

Gives the names of people who were sent the message directly.

All messages are in one of the following states:

deleted You used a **delete**, **dp**, or **dt** command to delete the message, or you saved it using a **Save** or **save** command and the variable *keepsave* was not set. When **mailx** quits, it deletes messages in this state.

new The message is in the system mailbox and you have not yet read it or otherwise changed its state. When **mailx** quits, it retains messages in this state in your system mailbox.

preserved

You used a **preserve** command on the message. When **mailx** quits, it retains messages in this state in their current locations.

read You used one of the following commands on the message:

~F	~m	next	print	type
~f	copy	pipe	top	undelete
~M	mbox	Print	Type	

or you used **delete**, **dp**, or **dt** on the preceding message and the *autoprint* environment variable was set. When **mailx** quits and you are in your system mailbox, it saves *read* messages in your personal mailbox unless the variable *hold* is set, in which case, it retains them in your system mailbox. If you are in your personal or a secondary mailbox when **mailx** quits, it retains *read* messages in their current location.

unread You have run more than one **mailx** session with the message in the system mailbox and you have not read it or otherwise changed its state. When **mailx** quits, it retains messages in this state in your system mailbox.

Command Mode Commands

The format of a command mode command is

```
[ command ] [ refs ] [ arguments ]
```

If no **command** is specified, **p[rint]** is assumed.

The *refs* argument indicates the messages to which you want to apply the **command**. **mailx** numbers incoming messages sequentially as they are received. The easiest way to refer to a message is to give its number. For example, the command

```
p 3
```

displays message number 3. At any point in a **mailx** session, there is one message that is considered the *current message*. Loosely speaking, this is the message you most recently used in a command (for example, the one you most recently read). If you omit the *refs* argument in a command that uses *refs*, the command works with the current message. You can also use special notations as the *refs* value, as in Table 1-7, *Reference Notations*.

Ref	Meaning
n	Message number n
n-m	Messages n through m
.	The current message
^	The first undeleted message (or first deleted message for undelete)
\$	The last message
*	All messages
+	Next message
-	Previous message
user	All messages from the given user
/string	All messages with <i>string</i> in the subject line (the case of characters in <i>string</i> is ignored)
:d	All deleted messages
:n	All new messages
:o	All old messages
:r	All messages that have already been read
:u	All unread messages

Table 1-7: Reference Notations

Several *refs* may be specified for the same command, separated by spaces. For example,

```
p alice lewis
```

displays all messages from *alice* plus all messages from *lewis*.

The *arguments* allowed at the end of a command mode command depend on the command itself. If a command allows a file name as an argument, the usual file name generation characters may be used in the file name. See **sh**(1).

The following list shows the commands recognized in command mode. In every command name, some characters are enclosed in square brackets. These characters are optional. For example, the **p[rint]** command may be given as **print** or **p**.

a[lias] [*alias* [*name* ...]]

sets up an address *alias*. If you issue a command to send mail to the given *alias*, the messages are actually sent to the given list of names. For example, you might issue the command

```
alias joe jsmith
```

From this point onward, you can address messages to *joe* and they are sent to *jsmith*. You may also set up an alias for several people, as in

```
alias choir soprano alto tenor bass
```

Once you have done this, you can send messages to `choir` and they are sent to the names that follow `choir` in the command. Issuing the **alias** command without any arguments displays a list of the currently defined aliases.

Note: Aliases which are entered interactively remain in effect only until the end of the current interactive session. To make an alias permanent, include the **alias** command in your start-up file (see *Start-Up Files*). See also **group**.

alt[ernates] *name ...*

lists a set of alternate names for your own login name. This is useful for people who login under several different names. When you **reply** to a message, **mailx** usually sends your reply to the author of the message and all the recipients as well; however, it does not send the message to any of your alternate login names. In this way, you don't have to worry about sending mail to yourself.

Specifying **alternates** without *names* displays your list of currently defined alternate names.

cd *directory*

makes *directory* your new working directory. If no *directory* is specified, **cd** goes to your *HOME* directory.

ch[dir] *directory*

is the same as **cd**.

c[opy] [*filename*]

copies the current message into the specified file. If the file does not already exist, it is created. If no *filename* is specified, your *mbox* file is used.

This operation does not mark the message as *saved*; if it was previously unread, it is still regarded as an unread message. Thus the original message remains in your system *mailbox*. See also **save**.

c[opy] *refs filename*

copies the messages referenced by *refs* into the given file. The *filename* must be specified. If the file does not already exist, it is created. As with the previous form of **copy**, the messages are not marked as *saved*.

C[opy] [*refs*]

is similar to the **copy** command, except that the referenced messages are saved in a file the name of which is derived from the author of the first message referenced. The name of the file is the author's name, stripped of any network addressing. If the *folder* variable is set, the file is saved to the specified directory. The copied messages are not marked as *saved*. If no *refs* are specified, the current message is copied.

d[delete] [*refs*]

deletes the specified messages. If no *refs* are specified, the current message is deleted. After a delete operation, the current message is set to the message after the last message deleted. Deleted messages are not thrown away until you end your session with the current *mailbox* (see **quit**, **file**). Until then, they can be undeleted (see **undelete**).

di[scard] [*header ...*]

does not display the given *header* fields when displaying a message. For example,

```
discard References
```

tells **mailx** not to display the *References* line at the beginning of any mail message. These header lines are retained when the message is saved; they are just not shown when the message is displayed. See also **ignore** and **retain**.

dp [*refs*]

deletes the specified messages, then displays the message after the last message deleted. If there is no subsequent message, **mailx** displays its command prompt.

dt [*refs*]

is the same as the **dp** command.

ec[ho] *string ...*

echoes the given *strings* (like the **echo**(1) command).

e[dit] [*refs*]

lets you edit the messages specified by *refs*. The messages are stored in a temporary file and an editor is invoked to let you edit the file. The default editor is **ed**(1), but you can change this using the *EDITOR* environment variable (see the *ENVIRONMENT VARIABLES* section).

ex[it] quits **mailx** without changing the system *mailbox*. Contrast this with **quit**.

fi[le] [*filename*]

quits the current mailbox (as if a **q[uit]** command was executed), then reads in the specified file as the new mailbox to examine. If no *filename* is specified, **mailx** displays the name and status of your current mailbox.

Several special strings can be used in place of *filename*:

%	your system <i>mailbox</i>
%user	the system <i>mailbox</i> for user
#	the previous file
&	your <i>mbox</i>
+file	the named file in the <i>folder</i> directory

fold[er] [*filename*]

is the same as the **file** command.

folders displays the names of the files in the directory given by the *folder* variable; see the *ENVIRONMENT VARIABLES* section.

F[ollowup] [*refs*]

replies to the first message given in the *refs*; **mailx** sends this reply to the authors of every message given in the *refs*. The **Subject** line is taken from the first message in the *refs*. **mailx** automatically saves your reply in a file which derives its name from the author of the message to which you are replying.

To create your reply, **mailx** puts you into input mode, where you can use all of the input mode commands.

fo[llowup] [*ref*]

replies to the specified message; if no message *ref* is given, you reply to the current message. **mailx** automatically saves your reply in a file which derives its name from the author of the original message. This overrides the *record* environment variable if it is set; see the *ENVIRONMENT VARIABLES* section.

To create your reply, **mailx** puts you into input mode, where you can use all of the input mode commands.

f[rom] [*refs*]

displays the header summary for the specified messages. If no *refs* are given, the current message is used.

g[roup] [*alias* [*name* ...]]

is the same as the **alias** command.

h[eaders] [*ref*]

displays the headers of a screenful of messages including the message given by *ref*. The number of lines in a screen is given by the *screen* environment variable; see the *ENVIRONMENT VARIABLES* section.

hel[p] displays a summary of the command mode commands.

ho[ld] [*refs*]

retains the specified messages in your system *mailbox*. For example, you might decide to **hold** a message if you read it, but decide not to act upon it immediately. If no *refs* are specified, the current message is held. If any of the specified messages have been marked as deleted, the **hold** command overrides that and still retains the messages. Subsequent **delete**, **dp**, and **dt** commands during the same **mailx** session can delete files marked for retention. See also **preserve**, and the environment variables *hold* and *keepsave*.

i[f] *code*

mailx commands

[el[se]

mailx commands

en[dif] is primarily intended for use in start-up files; see the *Start-Up Files* section for information. The *code* must be the character *r* or *s*. If it is *r*, the first set of *mailx commands* are executed if **mailx** is in *receive mode*, and the second set if **mailx** is in *send mode*. If *code* is *s*, the opposite is true. The **else** part is optional.

ig[nore] [*header ...*]

is the same as the **discard** command.

l[ist] displays the names of all command mode commands.

m[ail] *address ...*

sends a message to the specified recipients. **mailx** goes into input mode to let you enter the text of the message.

mb[ox] [*refs*]

indicates that the given messages are to be saved in your *mbox* when **mailx** quits normally (that is, through the **quit** command as opposed to **exit**).

n[ext] [*refs*]

goes to the next message in the mailbox that appears in the list of *refs*. For example,

n *user*

goes to the next message from the specified *user*.

pi[pe] [[*refs*] *command*]

pipes the messages given by *refs* through the specified shell *command* (run by the command interpreter identified by *SHELL*) These messages are considered read. If no *refs* are given, the current message is used. If no *command* is given, **mailx** uses the command given by the *cmd* environment variable; see the *ENVIRONMENT VARIABLE* section. If the *page* variable has a value, a formfeed is sent into the pipe after every message. The command

| [*refs*] [*command*]

is equivalent to **pipe**.

pre[serve] [*refs*]

is the same as the **hold** command.

P[rint] [*refs*]

displays the specified messages on the screen. If no *refs* are given, the current message is displayed. All header fields are displayed; the **discard** and **ignore** commands do not affect **Print**.

p[rint] [*refs*]

displays the specified messages on the screen. If no *refs* are given the current message is displayed. Header fields specified by **discard** and **ignore** commands are not displayed. If the *crt* variable is set to an integer, messages with more lines than that integer are *paginated* using the command specified by the *PAGER* variable. For more information, see the *ENVIRONMENT VARIABLES* section.

q[uit]

terminates a **mailx** session. This is the usual method to leave **mailx**. Messages that have been read but not saved or deleted are stored in your *mbx*. Messages that are still unread are retained in your system *mailbox*. Messages that have been deleted or explicitly saved in other files are discarded. Typing the EOF character has the same effect.

R[eply] [*refs*]

sends a reply to the authors of each of the messages specified by *refs*. If no *refs* are specified, the current message is used. The *Subject* line of the reply message is taken from the first message in *refs*. If the *record* environment variable is set to a file name, your reply message is appended to the end of that file.

Normally, you use **Reply** if you just want to send your reply to the author of a message, and **reply** if you want to send your reply to the author and all recipients. If set, the *flipr* variable reverses the meanings of the **R** and **r** commands. See the *ENVIRONMENT VARIABLES* section.

r[eply] [*ref*]

sends a reply to the author of a specific message, and all other recipients of the message. If no *ref* is specified, **mailx** replies to the current message. If the *record* environment variable is set to a file name, your reply message is appended to the end of that file.

R[espond] [*refs*]

is the same as the **Reply** command.

r[espond] [*ref*]

is the same as the **reply** command.

ret[ain] [*header ...*]

is the opposite of the **discard** command. It tells **mailx** to display the given *header* fields when displaying a message. The comparison of *header* fields is not case sensitive. You can use **retain** to override existing **discard** and **ignore** commands. If you do not specify any *header* fields, **retain** displays a list of currently retained header fields.

S[ave] [*refs*]

saves the specified messages in a file the name of which is taken from the author of the first message (the file name is the author's name, without any attached network addressing). If the *folder* variable is set, the file is saved to the specified directory.

s[ave] [*refs*] [*filename*]

saves the specified messages in the given file. If no *refs* are given, the current message is saved. The file is created if it doesn't already exist. If you do not specify *filename*, **mailx** saves the messages in *mbox*. A message that has been saved with **save** is normally deleted from *mailbox* when **mailx** terminates (see **quit**); but see the variables *hold* and *keepsave*.

se[t] [*name*]

defines a variable with the given *name* and assigns it a null value. If you omit *name*, **set** displays a list of all defined variables and their values.

se[t] *name=value*

defines a variable with the given *name* and assigns it the given value. The *value* may be a string or a number.

se[t] **noname**

is the same as the **unset** *name* command.

sh[ell] invokes the shell given by the *SHELL* environment variable.

si[ze] [*refs*]

displays the size in bytes of each of the specified messages. If no *refs* are specified, the current message is used.

so[urce] *file*

reads the specified text *file*, executes its contents as command mode commands, then returns to read more commands from the original source.

to[p] [*refs*]

displays the first few lines of each of the specified messages. If no *refs* are specified, the current message is used. If the *toplines* variable has a numeric value, then a number of lines equal to this value are displayed from each message; otherwise, five lines are displayed from each message.

tou[ch] [*refs*]

touches the specified messages, making them appear to have been read. This means that when you **quit mailx**, the messages are saved in your *mbox* if they are not deleted or explicitly saved in another file. If no *refs* are specified, the current message is touched.

T[ype] [*refs*]

is the same as the **Print** command.

t[ype] [*refs*]

is the same as the **print** command.

una[lias] [*alias [name ...]*]

deletes specified alias names.

u[ndelete] [*refs*]

restores previously deleted messages. When messages are deleted, they are not discarded immediately; they are just marked for deletion, and are actually deleted when **mailx** terminates. Until termination, you can use **undelete** to restore the specified messages. You cannot **undelete** messages deleted in previous sessions. If you do not give *refs*, this command restores the first deleted (but not yet undeleted) message following the current message; if no such message exists, it restores the last deleted (but not yet undeleted) message preceding the current message. If the *autoprint* variable is set, the last restored message is printed. This is the only command that lets you give a *ref* to a message which has been deleted.

U[nread] [*refs*]

marks the specified messages as unread.

uns[et] *name ...*

discards the specified variables.

ve[rsion]

displays version information about **mailx**.

v[isual] [*refs*]

edits the specified messages with a screen editor. If no *refs* are specified, the current message is edited. The messages are saved in a temporary file and the screen editor is invoked to edit that file. The editor used is given by the *VISUAL* variable; see the *ENVIRONMENT VARIABLES* section.

w[rite] [*refs filename*]

writes the specified messages into the given file. If no *refs* are given, the current message is written. **write** is the same as **save** except that it does not write out the header lines and the blank line at the end of the message.

x[it] is the same as the **exit** command.**z[+]** scrolls the header display forward one screenful.**z-** scrolls the header display backward one screenful.

! *command*

executes the given shell *command*. For example,

```
!lc
```

lists all files in the current directory. The shell that is invoked to execute the command is given by the *SHELL* environment variable; see the *ENVIRONMENT VARIABLES* section.

#*comment*

mailx ignores everything from the # to the end of the line. This is useful for putting comments into start-up files.

? displays a summary of command mode commands.

= displays the current message number.

Input Mode Commands

You may use input mode commands when entering the text of a message. Input mode commands must appear at the beginning of an input line; they cannot be in the middle of a line. By default, each input mode command begins with the tilde (~) character, called the *escape character*. You may use the *escape* environment variable to change the escape character, but the documentation that follows uses tilde.

~A inserts the *autograph string* at this point in the message. This autograph string is given by the *Sign* environment variable.

~a is similar to **~A**, except that it uses the variable *sign*.

~b *name* ...
adds the specified names to the blind carbon copy list.

~c *name* ...
adds the specified names to the carbon copy list.

~d reads in the *dead.letter* file; see the description of *DEAD* in the *ENVIRONMENT VARIABLES* section.

~e invokes an editor on the message that you have composed. The *EDITOR* variable determines the editor that is invoked.

~F [*refs*]
forwards the given messages. The text of the messages is inserted at this point in the message that you are composing. The message headers are also inserted with all header fields regardless of the **discard**, **ignore**, and **retain** commands. This is only valid when you entered **mailx** in command mode, then went into input mode to compose a message.

-
- ~f** [*refs*] is similar to **~F** except that the header fields included are determined by the **discard**, **ignore**, and **retain** commands.
- ~h** prompts you to enter the following header lines:
- ```
Subject Cc Bcc To
```
- For some of these, **mailx** displays an initial value for the header. You can edit this initial value as if you had just typed it in yourself, using backspaces and line deletes.
- ~i** *name* inserts the value of the named variable followed by a newline at this point in the message.
- ~M** [*refs*] inserts the text of the specified messages at this point in the message. If no *refs* are specified, the current message is used. Messages inserted in this way have each line prefixed with the value of the *indentprefix* variable. The message headers are also inserted with all header fields included regardless of the **discard**, **ignore**, and **retain** commands. This is only valid when you entered **mailx** in command mode, then went into input mode to reply to a message.
- ~m** is similar to **~M** except that the header fields included are determined by the **discard**, **ignore**, and **retain** commands.
- ~p** displays the message being composed.
- ~q** quits input mode as if you had interrupted the message. If you have already composed part of a message, the partial message is saved in the *dead.letter* file; see the description of the *DEAD* environment variable for more information.
- ~r** *filename* reads in the contents of the specified file and adds that text at this point in the message.
- ~s** *text* sets the Subject line to the given *text*.
- ~t** *address address ...* adds the given addresses to the To: list (people who will receive the message).
- ~v** invokes a screen (visual) editor on the message that you have composed. The *VISUAL* variable determines the editor that is invoked.
- ~w** *file* writes the current text of your message to the specified *file*. The header lines for the message are not written.
- ~x** quits in the same way as **~q**, except that the message is not saved in the *dead.letter* file.

`~<filename`

is the same as the `~r` command.

`~<!command`

executes the given shell *command* and adds the standard output of that command at this point in the message. For example, your message might contain

```
My program is giving me this odd output:
~< !prog
What do you think is causing it?
```

`~|command`

pipes the current message through the specified shell *command*. If the *command* terminates with a successful exit status, the output of the command replaces the text of the current message. For example,

```
~|fmt
```

fills and justifies the lines of your message and replaces the message with the formatted message. `~|` uses the shell given by the *SHELL* environment variable to run *command*.

`~!command`

executes the given shell *command*. For example,

```
~! lc
```

can be used to obtain a list of files in the current directory. The shell that is invoked to execute the command is given by the *SHELL* environment variable; see the *ENVIRONMENT VARIABLES* section. If the *bang* variable is set, **mailx** replaces each unescaped exclamation mark (!) in *command* with the command executed by the previous ! command or `~!` command escape.

`~.` marks the end of input in a mail message.

`~:mail_command`

executes the given command mode *mail\_command*. This is only valid when you entered **mailx** in command mode, then went into input mode to compose a message.

`~_mail_command`

is the same as the `~:` command.

`~?` displays a summary of the input mode commands.

## Start-Up Files

When you invoke **mailx** in command mode, **mailx** does the following:

1. Sets all variables to their default values. Processes command-line options, using them to override any corresponding default values.
2. Imports appropriate external environment variables, using them to override any corresponding default values.
3. Reads commands from the *system start-up file*, `/etc/mailx.rc`. This sets up variable values and definitions that should be common to all users. If you do not want **mailx** to read the system start-up file, use the `-n` option on the **mailx** command line.
4. Reads your *personal start-up file*, given by the environment variable `MAILRC`. By default, this is the `.mailrc` under your home directory.

Typically, start-up files set up display options and define aliases; however, any command is valid in a start-up file except for the following:

|        |         |          |          |
|--------|---------|----------|----------|
| Copy   | edit    | followup | Followup |
| mail   | hold    | preserve | reply    |
| Reply  | respond | Respond  | shell    |
| visual | !       |          |          |

If a line in a start-up file contains an error or an invalid command, the rest of the start-up file is ignored. **mailx** ignores blank lines in a start-up file.

## EXAMPLES

The following example composes and sends a message to several users. Items shown in italics are output by **mailx** itself.

```
mailx jean
Subject: Greetings
This is just a short note to say hello.
~c juan john johann
~.
```

On the first line, the message is just addressed to jean. The `~c` line adds more people who will receive copies of the message.

## ENVIRONMENT VARIABLES

A large number of variables control the behavior of **mailx**. These environment variables are divided into two classes: ones which always come from the external environment; and ones that may be set up in either the external environment or within a **mailx** session.

---

The following variables always come from the external environment; these can be changed inside a **mailx** session, except where marked.

**HOME**

gives the name of your home directory. This cannot be changed inside **mailx**.

**LOGNAME**

gives your login name.

**MAILDIR**

gives the name of the directory where system mailboxes are stored. If this is not set, the default is `/usr/mail`. The actual name of a user's system mailbox is derived in a system-dependent way by combining **MAILDIR** and the user's login name. For **mailx** to work properly, the **MAILDIR** directory must exist.

**MAILRC**

gives the name of your start-up file. This cannot be changed inside **mailx**. By default, **MAILRC** has the value `$HOME/.mailrc`. For more on start-up files, see the *Start-Up Files* section.

The **HOME** and **LOGNAME** variables *must* be set before you enter **mailx**; otherwise, **mailx** does not work properly. These variables are set automatically for you if you login using the **login**(1) facilities. If you do not login, you must set the variables in some other way, using the commands:

```
export LOGNAME=name
export HOME=directory
```

The remaining variables may be set in the external environment or in the course of a **mailx** session. The value of a variable may be set or changed with the **set** command; a variable may be discarded with the **unset** command. You will find it convenient to create a start-up file that sets these variables according to your preferences; this eliminates the need to set variables by hand every time you enter **mailx**.

Many of the following variables represent on/off options. If you set the variable itself (to any value), the option is turned on. To turn the option off, you may **unset** the variable, or **set** a variable consisting of **no** followed by the name of the original variable. For example, setting *autoprint* turns the autoprint option on, and setting *noautoprint* turns it off.

**allnet**

assumes that network addresses with the same login component refer to the same person. Network addresses typically consist of several *components*, giving information that lets a mail server identify a machine on the network, a route to that machine, and the login name of a user on that machine. **mailx** assumes that the login name is the last component.



For example,

```
print name
```

displays all messages that originated from the same login name, regardless of the rest of the network address. The default is *noallnet*, where different addresses are assumed to be different users, even if the login name components are the same.

*append*

appends messages to the end of the *mbox* file upon termination. The default is *noappend*; messages are placed at the beginning of the *mbox* file instead of the end.

*ask*

prompts you for a Subject: line when composing a message (if you have not already specified one with the **-s** option). This option is turned on by default; to turn it off, set *noask*.

*askbcc*

prompts you for a Bcc: list when composing a message. The default is *noaskbcc*; you are not prompted.

*askcc*

prompts you for a Cc: list when composing a message. The default is *noaskcc*; you are not prompted.

*asksub*

is the same as *ask*. *noasksub* is the same as *noask*.

*autoprint*

displays the next message automatically when you **delete** a message, and displays messages as you **undelete** them. The default is *noautoprint*; you are not shown messages that you **delete** or **undelete**.

*bang*

records shell commands executed inside the **mailx** session (for example, through the **~!** input mode command). Then, if you issue a shell command and the shell command contains a **!** character, **mailx** replaces that character with the command line for the previous shell command. The default is *nobang*, in which case a **!** in a shell command line is not treated specially.

*cmd*

contains a command, possibly with options. This specifies a default command line to be used for the command mode **pipe** command. For example,

```
set cmd="more"
```

pipes messages through **more** when the **pipe** command is invoked.

*conv*

specifies that UUCP network addresses are to be converted to a different style. The *conv* variable is assigned a code word indicating the desired style. At present, the only code word recognized is *internet*, which stands for the RFC822 specifications for network mail addressing. To make use of such addresses, you must have mail server software that can send and receive messages using such addresses; MPE/iX Shell and Utilities itself does not provide any software for transmitting messages to different systems. By default, *conv* is not defined and no conversion takes place.

*crt*

contains an integer number. If a message has more than this number of lines, **mailx** pipes the message through the command given by the *PAGER* variable, whenever it displays the message. If this variable is set to null, **mailx** treats it as a value of zero and pipes all messages through *PAGER*. The default is *nocrt*.

*DEAD*

contains the name of a file that can be used as the *dead.letter* file. Partial messages are saved in this file if an interrupt or error occurs during creation of the message or delivery. By default, the name of this file is *\$HOME/dead.letter*.

*dot*

accepts a line consisting of only a dot (.) as the end of a message in input mode (it is equivalent to ~.). The default is *nodot*. If *ignoreeof* is set, **mailx** ignores a setting of *nodot*; the dot is the only way to terminate input mode.

*EDITOR*

gives a command, possibly with options, that is invoked when using the command mode **edit** or the input mode ~e. The default is the **ed** utility (see **ed(1)**).

*escape*

gives the character used to begin input mode commands. The default is the tilde (~). If this variable is set to null, **mailx** disables command escaping.

*flipr*

reverses the meanings of the **R** and **r** commands. The default is *noflipr*. See also *Replyall*.

*folder*

contains the name of the directory in which **mailx** saves mail files. Whenever you specify a file name for a **mailx** command, putting a plus sign (+) in front of the name specifies that the file is to be accessed under the *folder* directory. If the value of *folder* begins with a slash, it is taken as an absolute path name; otherwise, **mailx** assumes that the directory is directly under your *HOME* directory. *folder* has no default value. If it is not set, the plus sign (+) has no special meaning at the beginning of file names.

*header*

displays a summary of message headers at the beginning of a **mailx** command mode session. This is the default.

*hold*

keeps all messages in your system *mailbox* instead of saving them in your *mbox*. The default is *nohold*.

*ignore*

ignores interrupts received while composing a message. The default is *noignore*.

*ignoreeof*

ignores EOF markers found while entering a message. The message can be ended by a dot (.) or ~, on a line by itself. The default is *noignoreeof*.

*indent*

contains a string that **mailx** uses as a prefix to each line in messages that ~**m** and ~**M** insert. The default is one TAB character.

*indentprefix*

the same as *indent*, contains a string that **mailx** uses as a prefix to each line in messages that ~**m** and ~**M** insert. The default is one TAB character. If both *indent* and *indentprefix* are set, *indentprefix* takes precedence.

*keep*

does not remove your system *mailbox* if the mailbox contains no messages. The *mailbox* is truncated to zero length. If the default *nokeep* is in effect, empty mailboxes are removed.

*keepsave*

keeps messages in your system *mailbox* even if they have been saved in other files. The default, *nokeepsave*, deletes messages from the *mailbox* if they have been saved elsewhere.

*LISTER*

contains a command, possibly with options. **mailx** invokes this command when displaying the contents of the *folder* directory for the **folders** command. If this variable is null or unset, **mailx** uses the **ls** utility. By default, this variable is unset.

*MAILRC*

location of personal start-up file. See *Start-Up Files*.

*MBOX*

gives the name of your *mbox* file. Messages that have been read but not saved elsewhere are saved here when you **quit** (but not when you **exit**). The default is *\$HOME/mbox*.

*metoo*

when replying to a message with your login name in the recipient list, sends a reply to all other recipients, the author and you. If *nometoo* is set, you are not sent the reply. The default is *nometoo*.

*onehop*

attempts to send replies directly to the recipients instead of going through the original author's machine. When you reply to a message, your reply is sent to the author and all recipients of the message. On a network, **mailx** normally specifies the recipient addresses so that the replies all go to the original author's machine first, then on to the other recipients.

*outfolder*

causes files used to record outgoing messages (see the description of *record*) to be located in the directory given by *folder* unless *folder* contains an absolute path name. The default is *nooutfolder*.

*page*

tells the **pipe** command to insert a formfeed character after each message that it sends through the pipe. The default is *nopage*.

*PAGER*

contains a command, possibly including options. **mailx** sends display output through this command if the output is longer than the screen length given by *screen*. The default is the **more** utility (see **more**(1)).

*prompt*

contains a string that **mailx** displays to prompt for output in command mode. The default is a question mark followed by a space (? ).

*quiet*

does not display the opening message and version number when **mailx** begins a session. The default is *noquiet*.

*record*

contains a file name where **mailx** records every message you send. If *record* is not an absolute path name and the *outfolder* variable has not been set, the file is located under the *HOME* directory. If the *outfolder* variable is set, the file is located in your *folder* directory. The default is *norecord*.

*Replyall*

reverses the senses of the **reply** and **Reply** commands (so that **reply** only replies to the author of a message, and **Reply** replies to the author and all other recipients). See also *flipr*.

*save*

saves messages in your *dead.letter* file if they are interrupted while being composed. The name of your *dead.letter* file is given by the *DEAD* variable. Setting *nosave* disables this automatic save feature. The default is *save*.

*screen*

gives the number of headers that are to be displayed by the **headers** and **z** commands.

*sendmail*

contains a command, possibly with options, that **mailx** invokes to send mail. The default is **mail**. It can be any command that takes addresses on the command line and message contents on standard input.

*sendwait*

when sending a message through a network, **mailx** waits for the mail server to finish before returning to your session. Normally, it just submits the message to the server, then returns immediately. The default is *nosendwait*.

*SHELL*

contains a command, possibly with options. **mailx** assumes that this command is a command interpreter. **mailx** invokes this command interpreter whenever it is asked to execute a system command (for example, through the **!** command mode command). The default is the MPE/iX Shell (see **sh**(1)).

*showto*

when displaying a header summary, displays the recipient's name instead of the author's for messages where you are the author. The default is *noshowto*.

*sign*

contains a string that is inserted into a message when you use the input mode **~a** command. **mailx** interprets **\n** and **\t** in this string as the newline and tab characters, respectively. The default is *nosign*.

*Sign*

contains a string that is inserted into a message when you use the input mode **~A** command. The default is *noSign*.

*toplines*

gives the number of header lines that the **top** command displays. The default is five.

*TERM*

contains the name of the terminal type. If *screen* is not set, *TERM* individually determines the number of lines in a screenful of headers.

*VISUAL*

contains a command, possibly with options, that **mailx** invokes when using the command mode **visual** or the input mode `~v`. The default is the **vi** utility (see **vi(1)**).

**FILES**

**mailx** uses the following files:

`/etc/mailx.rc`

System-wide start-up file.

*\$MAILRC*

Personal start-up file. By default, *MAILRC* has the value *\$HOME/.mailrc*.

*\$HOME/mbox*

Default location to save read messages. You can choose a different file by assigning the file name to the environment variable *MBOX*.

*\$MAILDIR*

Directory containing system mailboxes. By default, this is `/usr/mail`. You must create the *MAILDIR* directory if it does not already exist. For `/usr/mail`, you can create the directory with the commands

```
mkdir /usr
mkdir /usr/mail
```

if the directory doesn't already exist.

`dead.letter`

Default location to save partial letters.

**DIAGNOSTICS**

Possible exit status values are:

- 0 Successful completion. Also returned if **-e** is specified and there is new or unread mail.
- 1 Returned if **-e** is specified and there is no new or unread mail. Also returned to indicate failure because of any of the following:
  - there is no mail to read
  - inability to create temporary file name or temporary file
  - receipt of user interrupt while composing message
  - inability to determine the user's identity
- 2 Failure due to any of the following:
  - missing *number* after **-h**
  - missing *address* after **-r**
  - missing *subject* after **-s**
  - missing *user* after **-u**

- invalid command line option
- use of interactive options when not using command interactively

## Messages

- Message:** alias storage: *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** allocating message address: *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** allocating message header *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** At EOF  
**Cause:** You tried to advance past the last message in the mailbox.  
**Action:** Do not issue commands that try to access past the last message.
- Message:** cannot find out who you are  
**Cause:** The **mailx** command was unable to find your user ID.  
**Action:** Check with your system manager.
- Message:** Cannot "preserve" in edit mode  
**Cause:** You can only use the **preserve** or **hold** commands when working with your system mailbox.  
**Action:** Make sure you are working with your system mailbox if you want to use the **preserve** or **hold** commands.
- Message:** Command "*command*" is illegal in startup file.  
**Cause:** Your start-up file contains a command that cannot be used in such files.  
**Action:** See the *Start-Up Files* subsection for a list of commands that are not valid in start-up files. Remove the offending command from your start-up file.
- Message:** command "*command*": *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** command file "*cmdfile name*": *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** creating temporary file name: *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).

- 
- Message:** else: no matching "if" statement  
**Cause:** You issued an **else** command without a corresponding **if** command.  
**Action:** Ensure that all **else** commands are preceded by a **if** command.
- Message:** endif: no matching "if" statement  
**Cause:** You issued an **endif** command without a corresponding **if** command.  
**Action:** Ensure that all **endif** commands are preceded by a **if** command.
- Message:** EOF inside "if" statement  
**Cause:** While processing an **if** command, **mailx** encountered an end-of-file condition.  
**Action:** If the **if** command is in your start-up file, ensure that you have included a corresponding **endif** command. If you are entering the **if** in command mode, do not enter the EOF character before issuing the **endif** command.
- Message:** Extra arguments after "*command*" command  
**Cause:** You included more arguments than *command* handles.  
**Action:** Check the *Command Mode Commands* subsection for the correct arguments for *command*.
- Message:** failed  
**Cause:** You tried to pipe output through a command and the command failed.  
**Action:** Verify that the command being used is a valid MPE/iX Shell and Utilities command that is capable of reading from the standard input.
- Message:** file "*filename*": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** fork failed: *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** fork for shell command: *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** fork for shell: *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** if: missing argument  
**Cause:** You did not provide an argument in an **if** command.  
**Action:** Provide either **r** or **s** as the argument of the **if** command.



- 
- Message:** if: "s" or "r" are permissible arguments  
**Cause:** You used an argument other than `r` or `s` with the `if` command.  
**Action:** Use only `r` or `s` as the argument for an `if` command.
- Message:** Inappropriate message  
**Cause:** You tried to perform a command on an inappropriate message. For example, you tried to undelete a message that was not deleted or you tried to respond to a deleted message.  
**Action:** Check the description of the command you are using to ensure that you are using it correctly.
- Message:** Invalid hop count: *num*  
**Cause:** You specified the `-h` option, but the argument that you provided with it is not a valid number.  
**Action:** Provide a valid number as the argument to the `-h` option
- Message:** Invalid message number  
**Cause:** You used a message number of 0 or one that is greater than the number of messages in the mailbox.  
**Action:** Use a message number in the range from 1 to the number of messages in the mailbox.
- Message:** mail file "*file*" is locked  
**Cause:** You tried to access a mailbox that another process was already accessing.  
**Action:** Wait until the other process completes its access then try again.
- Message:** mail to command "*command name*": *system error*  
**Cause:** See `syserror(3)`.  
**Action:** See `syserror(3)`.
- Message:** mail to file '*filename*': *system error*  
**Cause:** See `syserror(3)`.  
**Action:** See `syserror(3)`.
- Message:** Mismatched *quote char*  
**Cause:** You specified the quote character *quote char*. This quote character must be paired without a matching mate. The mate is missing.  
**Action:** Provide the missing mate to *quote char*.
- Message:** Missing address after "`-r`"  
**Cause:** You specified the `-r` option without providing an address as its argument.  
**Action:** Provide an address following the `-r` option.
- Message:** Missing file after `source` command  
**Cause:** You issued a `source` without specifying a file name.  
**Action:** Specify a file name with the `source` command.

- 
- Message:** Missing file name  
**Cause:** You issued a command which requires a file name without providing one.  
**Action:** Specify a file name.
- Message:** Missing number of hops after "-h"  
**Cause:** You invoked **mailx** with the **-h** option without providing a value following it.  
**Action:** Provide a value following the **-h** option.
- Message:** Missing subject after "-s"  
**Cause:** You invoked **mailx** with the **-s** option without providing a subject string following it.  
**Action:** Provide a subject string following the **-s** option.
- Message:** Missing user after "-u"  
**Cause:** You invoked **mailx** with the **-u** option without providing a user name following it.  
**Action:** Provide a user name following the **-u** option.
- Message:** No applicable messages  
**Cause:** You specified a message list that does not match any messages in the current mailbox.  
**Action:** No action required.
- Message:** No messages satisfy *:colon modifier*  
**Cause:** You specified a *colon modifier* (one of :d, :n, :o, :r, or :u) as your message list and none of the messages in the current mailbox matched it.  
**Action:** No action required
- Message:** No previous file.  
**Cause:** You used # to represent the file name of the previous file when there was no previous file.  
**Action:** Use a different file name indicator.
- Message:** No value set for *folder variable*"  
**Cause:** You have not provided a value for the **mailx** variable *folder*.  
**Action:** Provide a value for the variable *folder* either in the start-up file or in command mode.
- Message:** Non-numeric second argument  
**Cause:** The second argument in a message list was not numeric.  
**Action:** Ensure that, when specifying a range of messages as arguments for a command, you indicate the first and last message in the range with integers in the range 1 to the number of messages in the current mailbox.

- 
- Message:** not in interactive mode  
**Cause:** You attempted to use an interactive mode command (through ~:) when you did not invoke **mailx** in interactive mode.  
**Action:** Invoke **mailx** in interactive mode to use interactive mode commands.
- Message:** Option *-option* argument missing  
**Cause:** You did not provide an argument for *-option*.  
**Action:** Provide the missing argument.
- Message:** Options applying only to interactive use were given.  
**Cause:** You specified the **-e**, **-f**, **-H**, **-N**, or **-u** options when attempting to send mail. These options are only for use when reading mail.  
**Action:** Check the *Options* subsection for usable options when sending mail.
- Message:** *pathname : system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** read error: *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** Referencing before first message  
**Cause:** You used the - notation to try to reference the message before the first one in the mailbox.  
**Action:** Do not use - when the current message is the first message in the mailbox.
- Message:** Referencing beyond last message  
**Cause:** You used the + notation to try to reference the next message when the current message was the last one in the mailbox.  
**Action:** Do not use + when the current message is the last message in the mailbox.
- Message:** rewriting "*filename*" : *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** temporary file "*filename*" : *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** Too many mail folders specified on command line.  
**Cause:** When specifying the **-f** option, you named more than one mail folder on the command line.  
**Action:** List only one file name on the command line.

- 
- Message:** Unknown command "*command*"  
**Cause:** You entered a command that does that not exist in **mailx**.  
**Action:** Verify that you spelled the command name properly and that it is a valid **mailx** command.
- Message:** Unknown command "*~command*". Type *~?* for help.  
**Cause:** You entered an input mode command that does not exist.  
**Action:** Check the *Input Mode Commands* subsection to ensure that the command is valid. Remember that *~* is the default escape character and may be changed by the *escape* variable.
- Message:** Unknown option "*-option*"  
**Cause:** You specified a command line option that is not valid for **mailx**.  
**Action:** Check the *Options* subsection for a list of valid **mailx** options.
- Message:** Unknown colon modifier "*:colon modifier*"  
**Cause:** You used an unknown colon modifier in a message list.  
**Action:** Check the *Command Mode Commands* subsection for a list of valid colon modifiers.
- Message:** *variable: no such variable*  
**Cause:** You tried to make use of a variable that does not exist.  
**Action:** Check to make sure that you have spelled the variable name correctly or define the variable with a **set** command.
- Message:** *variable: read-only variable*  
**Cause:** You cannot change the values of some environment variables, such as *HOME* and *MAILRC*, from within **mailx**. You tried to change the value of such a variable.  
**Action:** Do not try to change the value of read-only variables.
- Message:** *variable "cmd" not set*  
**Cause:** You invoked the **pipe** command without any arguments and have not assigned a value to the *cmd* environment variable. This variable indicates the default command to pipe messages through.  
**Action:** Assign a value to *cmd*.
- Message:** *variable storage: system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** *write error: system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).

---

**PORTABILITY**

POSIX.2. *x*/OPEN Portability Guide 4.0. UNIX System V.

UNIX System V has a compatible **mailx** utility while Berkeley systems have a similar utility known as **Mail**.

The **-d**, **-F**, **-r**, and **-U** options, the **Copy**, **echo**, **followup**, **Followup**, **Save**, **Unread**, and **version** commands, and the *allnet*, *conv*, *onehop*, *replyall*, *sendmail*, and *sendwait* variables are extensions to the POSIX standard.

**MPE/iX NOTES**

Since the current MPE/iX implementation of **mailx** does not support the native MPE mailer, you cannot use it to send mail to other systems. On MPE/iX, it uses the **tmail** delivery agent.

Also, since an MPE/iX child process cannot survive the death of its parent, you should always use mailx with the variable *sendwait* turned on. By default, this variable is turned off; however, the default system configuration file for **mailx** (*/etc/mailx.rc*) contains a line that turns it on.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

**SEE ALSO**

**echo(1)**, **ed(1)**, **fmt(1)**, **sh(1)**, **vi(1)**, **tmail(3)**

**NAME**

make — maintain program-generated and interdependent files

**SYNOPSIS**

```
make [-Eeinpqrstuvvx] [-k | -S] [-c dir] [-f file] ...
[macro definition] ... [target ...]
```

**DESCRIPTION**

**make** is a command that helps you manage projects that contain a set of interdependent files. Typical examples would be a program with many source and object files, or a document that is built from source files, macro files, and so on. **make** keeps all the files up-to-date with one another: if one file changes, **make** updates all the other files that depend on the changed file.

**Note:** This implementation of **make** features the `.POSIX` special target to provide maximum portability. When you specify this target, **make** processes the makefile as specified in the POSIX.2 standard. For details, see the description of `.POSIX` in the *Special Targets* section of this man page.

**Options**

**make** accepts the following options:

- c** *dir* attempts to change into the specified directory when **make** starts up. If **make** can't change directory, an error message is displayed. This is useful for recursive makefiles when building in a different directory.
- E** suppresses reading of the environment. If neither **-E** nor **-e** are specified, **make** reads the environment *before* reading the makefile.
- e** reads the environment *after* reading the makefile. If neither **-E** nor **-e** are specified, **make** reads the environment *before* reading the makefile.
- f** *file* uses *file* as the source for the makefile description. **make** ignores the makefiles specified as prerequisites to the `.MAKEFILES` target. If you specify *file* as `-`, **make** reads the standard input.
- i** ignores all errors and continues making other targets. This is equivalent to the `.IGNORE` attribute or macro.
- k** makes all independent targets, even if an error occurs. Specifying **-k** tells **make** to ignore the error and continue to make as much as possible. **make** does not attempt to update anything that depends on the target that was being made when the error occurred.
- n** displays the commands that **make** would execute to update the chosen targets, but does not actually execute the commands unless they have a plus sign (+) prefix. When this option is specified, **make** displays lines with an at sign (@) prefix on the

---

standard output. This feature works with group recipes, but in these cases, **make** executes the commands. If **make** finds the string `$(MAKE)` in a recipe line, it expands it, adds `-n` to the `MAKEFLAGS`, and then executes the recipe line. This allows you to see what recursive calls to **make** do. The output correctly shows line breaks in recipes that are divided into several lines of text using the `\<newline>` sequence.

- `-p` displays the digested makefile including the complete set of macro and target definitions in a human readable form that is useful for debugging, but cannot be used as input to **make**.
- `-q` checks whether the target is up-to-date. If it is up-to-date, **make** exits with a status of 0; otherwise, it exits with a status of 1 (typically interpreted as an error by other software). When you specify `-q`, **make** does not execute any commands unless they have a plus sign (+) prefix.
- `-r` does not read the default rules from the startup file defined by the `MAKESTARTUP` environment variable.
- `-S` terminates **make** if an error occurs during operations to bring a target up-to-date (opposite of `-k`). This is the default.
- `-s` does not display the commands **make** executes, warning messages, or touch messages (see the `-t` option). This is equivalent to the `.SILENT` attribute or macro.
- `-t` touches the targets to mark them as up-to-date, but only executes commands to change a target if the target has a plus sign (+) prefix. **make** does not touch targets that are already up-to-date or targets that have prerequisites but do not have recipes. **make** displays a message for each target file which indicates the file name and that it was touched.
- `-u` forces an unconditional update: **make** behaves as if all the prerequisites of the given target are out-of-date.
- `-V` displays the version number of **make** and a list of built-in rules.
- `-v` displays a detailed account of **make**'s progress. This includes what files it reads, the definition and redefinition of each macro, meta-rule and suffix rule searches and other information.
- `-x` exports all macro definitions to the environment. This happens before **make** begins making targets (but after it reads the entire makefile).

### Targets

A *target* is normally a file that you want to ensure is up-to-date with the files on which it is dependent. For example, you may want to check to see if a compiled C program is based on the most recent version of its source code and if not, recompile the source code to get an

up-to-date version. In this case, the compiled program file is the target and the corresponding source code files are *prerequisites* (that is, the files on which a target is dependent).

**make** updates all targets that are specified on the command line. If you do not specify any target, **make** updates the targets in the first rule of the makefile. A target is out-of-date if it is older than any of its prerequisites (based on modification times) or if it does not exist. To update a target, **make** first recursively ensures that all the target's prerequisites are up-to-date, processing them in the order in which they appear in the rule. If the target itself is out-of-date, **make** then executes the recipe associated with the target. If the target has no associated recipe, **make** considers it up-to-date.

**make** also supports another form of targets known as *special targets* described in the *Special Targets* section of this man page.

### Makefiles

A *makefile* is a text file which describes the dependencies between various files. A makefile normally contains a list of targets and identifies the prerequisites on which each depends. It also contains a series of instructions, called *recipes* which describe the actions to be taken if a given target is out-of-date with its prerequisites.

By default, if you do not specify the **-f** option, **make** looks for a file in your current directory named `makefile`. If **make** does not find this file, it searches your current directory for a file named `Makefile`. If **make** finds either file, it is used as your makefile.

You can change the default makefiles with the `.MAKEFILES` special target (see the *Special Targets* section of this man page).

### Macro Definitions

Macro definitions may take several forms.

*macro* = *string*

is the usual form. If *string* contains macro references, **make** does not expand them when the macro is defined, but when the macro is actually used.

*macro* := *string*

expands macros inside *string* before creating *macro*.

*macro* += *string*

adds *string* to the previous value of *macro*.

You can use any amount of white space on both sides of macro operators. **make** defines the name *macro* to have the value *string* and replaces it with that value whenever it is used as `$(macro)` or `${macro}` within the makefile. It is possible to specify a `$(macro_name)` or `${macro_name}` macro expansion where *macro\_name* contains more `$(...)` or `${...}` macro expansions itself.



Normally, **make** does not include white space at the beginning and end of *string* in the definition of *macro*; however, it never strips white space from macros imported from the environment. If you want to include white space in a macro definition specified on the **make** command line, you must enclose the definition in quotes.

**make** resolves macro definitions in the following order:

1. Macro definitions in the built-in inference rules.
2. Contents of the environment.
3. Macro definitions in the makefiles (in the order they appear).
4. Macro definitions on the command line.

If a macro is already defined when **make** encounters a new definition for it, the new definition replaces the old one. For example, a macro definition for *name* on the command line overrides a definition for *name* in the makefile.

### Macro Modifiers

MAKE supports macro expansions of the form:

```
$(macro_name:modifier_list:modifier_list:...)
```

Possible modifiers are:

|                            |                                                  |
|----------------------------|--------------------------------------------------|
| <code>^ "string"</code>    | prefix tokens                                    |
| <code>+ "string"</code>    | suffix tokens                                    |
| <code>b</code>             | file portion of all path names, without suffix   |
| <code>d</code>             | directory portion of all path names              |
| <code>f</code>             | file portion of all path names, including suffix |
| <code>l</code>             | all characters mapped to lowercase               |
| <code>s/pat/string/</code> | simple pattern substitution                      |
| <code>suffix=string</code> | suffix replacement                               |
| <code>t "separator"</code> | tokenization                                     |
| <code>u</code>             | all characters mapped to uppercase               |

You may specify macro modifiers in either upper or lowercase.

For example, with

```
test = D1/D2/d3/a.out f.out d1/k.out
```

we have

```
$(test:d) -> D1/D2/d3 . d1
$(test:b) -> a f k
$(test:f) -> a.out f.out k.out
${test:DB} -> D1/D2/d3/a f d1/k
${test:s/out/in} -> D1/D2/d3/a.in f.in d1/k.in
$(test:f:t"+") -> a.out+f.out+k.out
$(test:t"+") -> D1/D2/d3/a.out+f.out+d1/k.out
$(test:u) -> D1/D2/D3/A.OUT F.OUT D1/K.OUT
$(test:l) -> d1/d2/d3/a.out f.out d1/k.out
$(test:~/rd/") -> /rd/D1/D2/d3/a.out /rd/f.out /rd/d1/k.out
$(test:+" .Z") -> D1/D2/d3/a.out.Z f.out.Z d1/k.out.Z
```

## Run-time Macros

Run-time macros can take on different values for each target.

**\$@** The full target name. When building a normal target, this macro evaluates to the full name of the target. When building a library, it expands to the name of the archive library. For example, if the target is

```
mylib(member)
```

**\$@** expands to

```
mylib.
```

**\$%** The full target name. When building a normal target, this macro evaluates to the full name of the target. When building a library, it expands to the name of the archive member. For example, if the target is

```
mylib(member)
```

**\$%** expands to

```
member
```

**\$&** The list of all prerequisites, in all rules that apply to the target. In `::` rules, this macro produces a value identical to the **\$\$** macro.

**\$\$** The list of all prerequisites which are newer than the target.

**\$\$** The list of all prerequisites taken from the list specified on the rule line of the recipe where the **\$\$** appears.

`$(` Subset of `^` which prompt the rule's execution (specified on the rule line where the `$(` appears). In normal rules, this contains the list of all recently changed prerequisites. In inference rules, this contains the single prerequisite associated with the current, executing rule.

`$>` The name of the library if the current target is a library member.

`$*` The target name with no suffix (`$(%:db)`) or the value of the stem in a meta-rule.

The constructs `$$@`, `$$%`, `$$>`, and `$$*` yield meaningful results when placed in a prerequisite list as a dynamic prerequisite.

`$$@` stands for the target currently being made. The following two examples are equivalent:

```
fred : $$@.c
fred : fred.c
```

The construct may be modified:

```
fred.o : $$(@:b).c
```

If you are building a library, `$$%` stands for the name of the archive member being made. If you are building a normal target, `$$%` stands for the name of the target currently being made.

`$$*` stands for the name of the current target being made, but with no suffix.

If you are building a library, `$$>` stands for the name of the archive library being made. If you are not building a library, its use is invalid.

### Comments

Comments begin with the number sign (#) character and extend to the end-of-line. **make** discards all comment text.

### Makefile Contents

Inside makefiles, you can split long lines over several lines of text. To do this, put a backslash (\) at the very end of the line. You can use this technique to extend comments as well as recipe lines and macro definitions for example.

If a rule or macro definition must contain a # character, use \#; otherwise, **make** mistakes the # for the beginning of a comment. Also, `$$` stands for `$`.

File names that contain a colon (:) must always be enclosed in quotes (" "):

```
"a:target" : "a:prereq"
```

## Rules

The general format of a rule is

```
targets [attributes] ruleop [prerequisites] [; recipe]
{<tab> recipe}
```

The parts of the rule are described as follows.

*targets* one or more target names.

*attributes*

a list, possibly empty, of attributes to apply to the list of targets.

*ruleop* a separator string that separates the target names from the prerequisite names and may also affect the processing of the specified targets.

*prerequisites*

a list of zero or more names on which the specified targets depend.

*recipe*

may follow on the same line as the prerequisites, separated from them by a semicolon (;). If such a recipe is present, **make** takes it as the first in the list of recipe lines defining how to make the named targets. Additional recipe lines may follow the first line of the rule. Each such recipe line must begin with a tab character.

The possible rule operators are listed as follows.

*targets* : *prereqs*

simple rule definition. You can specify only one set of rules for making a target, except within meta-rules. In meta-rules, you can specify more than one recipe for making the target. If a target has more than one associated meta-rule, **make** uses the first meta-rule that matches.

*targets* :! *prereqs*

executes the recipe for the associated targets once for each recently changed prerequisite. Normally, the recipe is executed only once, for all recently changed prerequisites at the same time.

*targets* :^ *prereqs*

inserts the specified prerequisites before any other prerequisites already associated with the specified targets.

*targets* :- *prereqs*

clears the previous list of prerequisites before adding the new prerequisites.

targets :: prereqs

is used for multiple rules applying to the same target. Each rule can specify a different set of prerequisites with a different recipe for updating the target. If a target is out-of-date with respect to any of its prerequisites, **make** remakes the target using all the recipe lines associated with the rules that mention those prerequisites.

targets :| prereqs

is used in meta-rules. It tells **make** to treat each meta-dependency as an independent meta-rule. For example:

```
%o :| archive/%.c rcs/%.c /srcarc/RCS/%.c
 recipe...
```

is equivalent to

```
%o : archive/%.c
 recipe...
%.o : rcs/%.c
 recipe...
%.o : /srcarc/rcs/%.c
 recipe...
```

This operator is particularly useful for searching for RCS archives. If the *RCSPATH* variable used by RCS is defined as

```
archive/%f;rcs/%f;/srcarc/rcs/%f
```

then the meta-rule

```
% : $(RCSPATH:s/%f/%/:s;/ /)
 co -l $<
```

searches the path looking for an RCS file and checks it out.

You can follow the first line of a rule with any number of recipe lines. Each of these must begin with a tab character. You can follow the tab with -, @, + or all three. - indicates that **make** is to ignore non-zero exit values when it executes this recipe line. @ indicates that **make** is *not* to display the recipe line before executing it. + tells **make** to always execute this line, even when **-n**, **-q**, or **-t** is specified. This is particularly useful when calling **make** recursively. If the recursive **make** line is preceded by a +,

```
make -n
```

executes the recursive **make** but puts the *n* in the *MAKEFLAGS* variable. This allows you to see what the subsidiary makes do.

You can use a target that has prerequisites but no recipes to add the given prerequisites to that target's list of prerequisites.

---

Group recipes begin with `[` in the first non-white space position of a line, and end with `]` in the first non-white space position of a line. Recipe lines in a group recipe need not have a leading tab. **make** executes a group recipe by feeding it as a single unit to a shell. If you immediately follow the `[` at the beginning of a group recipe with one of `-`, `@` or `+`, they apply to the entire group in the same way that they apply to single recipe lines.

### Inference Rules

With inference rules, you can specify general rules for building files rather than creating a specific rule for each target.

MAKE provides two forms of inferences rule: suffix rules and meta-rules. It provides suffix rules for compatibility with older makefiles. Meta-rules are a more general technique than suffix rules for specifying **make**'s default behavior. They provide a superset of the functionality of suffix rules.

**make** uses the inference rules to infer how it can bring a target up to date. A list of inference rules defines the commands to be executed. The default `startup.mk` contains a set of inference rules for the most common targets. You can specify additional rules in the makefile.

When **make** finds no explicit target rule to update a target, it checks the inference rules. If **make** finds an applicable inference rule with an out of date prerequisite, it executes that rule's recipe. (See also the section describing the `.DEFAULT` special target).

### Suffix Rules

**make** treats targets that begin with a period and contain no slashes or percent signs as suffix rules. If there is only one period in the target, it is a single-suffix inference rule. Targets with two periods are double-suffix inference rules. Suffix rules do not have prerequisites but do have commands associated with them.

When **make** finds no explicit rule to update a target, it checks the suffix of that target (`.s1`) against the suffix rules. **make** examines a prerequisite based on the base name of the target with the second suffix (`.s2`) appended, and if the target is out-of-date with respect to this prerequisite, **make** executes the recipe for that inference rule.

Meta-rules take precedence over suffix rules.

If the target to be built does not contain a suffix and there is no rule for the target, **make** checks the single suffix inference rules. The single suffix inference rules define how to build a target if **make** finds a rule with one of the single suffixes appended. A rule with one suffix `.s2` defines how to build *target* from *target.s2*. **make** treats the other suffix (`.s1`) as null.

For a suffix rule to work, the component suffixes must appear in the prerequisite list of the `.SUFFIXES` special target. You turn off suffix rules by placing

```
.SUFFIXES :
```

in your makefile. This clears the prerequisites of the `.SUFFIXES` target which prevents the enaction of any suffix rules. The order that the suffixes appear in the `.SUFFIXES` rule determines the order in which **make** checks the suffix rules.

The following steps describe the search algorithm for suffix rules:

1. Extract the suffix from the target.
2. Is it in the `.SUFFIXES` list? If not, then quit the search.
3. If it is in the `.SUFFIXES` list, look for a double suffix rule that matches the target suffix.
4. If you find one, then extract the base name of the file, add on the second suffix and see if the resulting file exists. If it doesn't, then keep searching the double suffix rules. If it does exist, then use the recipe for this rule.
5. If no successful match is made, then the inference has failed.
6. If the target did not have a suffix, then check the single suffix rules in the order that the suffixes are specified in the `.SUFFIXES` target.
7. For each single suffix rule, add the suffix to the target name and see if the resulting file name exists.
8. If the file exists, then execute the recipe associated with that suffix rule. If the file doesn't exist, continue trying the rest of the single suffix rules. If no successful match is made, then the inference has failed.

MAKE also provides a special feature in the suffix rule mechanism for archive library handling. If you specify a suffix rule of the form

```
.a.suf:
 recipe
```

the rule matches any target having the `LIBRARYM` attribute set, regardless of what the actual suffix was. For example, if your makefile contains the rules

```
.SUFFIXES: .a .o
.a.o :
 echo adding $< to library $@
```

then if `mem.o` exists

```
make "mylib(mem.o)"
```

causes

```
 adding mem.o to library mylib
```

to be printed.

Refer to *Making Libraries* in the *User's Guide* for more information about libraries and the .LIBRARY and .LIBRARYM attributes.

### Meta-rules

Meta-rules have one target with a single percent symbol which matches an arbitrary string called the stem;  $A\%B$  matches any string which starts with prefix  $A$  and ends with suffix  $B$ .  $A$  and/or  $B$  may be null. The  $\%$  in a dependency stands for the stem.

The inference rule to update a target matching pattern  $p1\%s1$ , where  $p1$  and  $s1$  are prefix and suffix strings of the target, having a prerequisite  $p2\%s2$ , where  $\%$  is the stem from the target, is specified as a rule:

```
 p1%s1 : p2%s2 ; recipe...
```

Either the prefix or suffix string may be empty

With the internal macros you can specify general inference rules. If the target is out-of-date with respect to this prerequisite, **make** executes that inference rule's recipe.

### Transitive Closure

Meta-rules provide a mechanism which allows several meta-rules to chain together to eventually create the target.

This is called *transitive closure*. For example, suppose you have the following two meta-rules

```
 %.o : %.c
 ... rule body...
```

and

```
 %.c : %.y
 ... rule body ...
```

When you specify

```
 make file.o
```

**make** uses the first meta-rule to look for `file.c`. If it can't find an explicit rule to build `file.c`, it again looks through the meta-rules and finds the rule that tells it to look for `file.y`.



**make** considers each meta-rule only once when performing transitive closure to avoid a situation where it loops forever. For example, if you have the rule

```
% : %.c
 ... rule body ...
```

the command

```
make file
```

causes **make** to look for `file.c`. If the meta-rules were not restricted and `file.c` did not exist, then **make** would look for `file.c.c`, and then `file.c.c.c`, and so on. Because **make** uses each meta-rule only once, this can't happen.

**make** computes transitive closure once for each meta-rule head the first time the pattern matches a target. When transitive closure is computed, **make** adds all the computed rules to the rule set for that meta-rule head. For example, if you have the rules

```
% : %.o
 recipe 1...
%.o : %c
 recipe 2...
```

and you are making `file`, this target matches successfully against `%` causing **make** to compute transitive closure for `%`. As a result of this computation, a new rule is created:

```
% : %.c
 recipe 2...
 .REMOVE target recipe for %.o, if not .PRECIOUS
 recipe 1...
```

**make** executes this rule if `file.o` doesn't exist. When **make** finishes the computation for the rule head; it marks the rule head as *transitive closure computed*. Since **make** adds all possible new rules to the rule set the first time the computation is done, it is not necessary to do it again — nothing new is added. The term *transitive closure* is adapted from mathematical set theory.

**Note:** In set theory, if you have a set composed of pairs (a,b) and (b,c), then the set would be transitively closed if (a,c) is also in the set. This is exactly what **make** does: it adds (a,c) to the set of meta-rules if there are already rules (a,b) and (b,c) in the set.

The best way to understand how this works is to experiment with little **make** files with the `-v` option specified. This shows you in detail what rules are being searched, when transitive closure is calculated and what rules are added.

## Attributes

**make** defines several target attributes. Attributes may be assigned to a single target, a group of targets, or to all targets in the makefile. Attributes affect what **make** does when it needs to update a target. You can associate attributes with targets by specifying a rule of the following form:

```
attribute_list : target ...
```

This assigns the attributes in *attribute\_list* to the given targets. If you do not specify any targets, the attributes apply to every target in the makefile. You can also put attributes inside a normal rule, as in:

```
targets attribute_list : prerequisite ...
```

These are the recognized attributes:

**.EPILOG**

Insert shell epilog code when executing a group recipe associated with any target having this attribute set.

**.IGNORE**

Ignore an error when trying to make any target with this attribute set.

**.LIBRARY**

Target is a library.

**.LIBRARYM**

Target is a library member (cannot be set by the user).

**.PRECIOUS**

Do not remove this target under any circumstances. Any automatically inferred prerequisite inherits this attribute.

**.PROLOG**

Insert shell prolog code when executing a group recipe associated with any target having this attribute set.

**.SETDIR**

Change current working directory to specified directory when making associated targets. The syntax of this attribute is `.SETDIR=path`, where *path* is the path name of desired working directory. If *path* contains any colon (:) characters, the entire attribute string must be quoted, not just the path name.

**.SILENT**

Do not echo the recipe lines when making any target with this attribute set, and do not issue any warnings.

**.SYMBOL**

Target is an entry point into a module in a library (cannot be set by the user). This attribute is used only when searching a library for a target. Targets of the form `lib( (entry) )` have this attribute set automatically.

You can specify any attribute except `.LIBRARYM` and `.SYMBOL`. You can use any attribute with any target, including special targets.

**Special Targets**

*Special Targets* are called targets because they appear in the target position of rules; however, they are really keywords, not targets. The rules in which they appear are really *directives* which control the behavior of **make**.

The special target must be the only target in a special rule — you cannot list other normal or special targets.

Some special targets are affected by some attributes. Any special target can be given any attribute, but often the combination is meaningless and the attribute has no effect.

**.BRACEEXPAND**

This target may have no prerequisites and no recipes associated with it. If set, the target enables the outdated brace expansion feature used in older versions of **make**. Older **make**s would expand a construct of the following form, beginning with each token in the token list:

```
string1{token_list}string2
```

Older **make**s would append `string1` to the front of each token in the list, and `string2` to the end of each token in the list. A more productive means for achieving the same result with modern versions of **make** relies on macro expansion with prefix and suffix modifiers:

```
$(TOKEN_BASE:^"prefix":+"suffix")
```

Note that the double quotes are required. Brace expansion is an outdated feature available in past versions of **make** and future versions will dispense with it completely.

**.DEFAULT**

This target has no prerequisites, but it does have a recipe. If **make** can apply no other rule to produce a target, it uses this rule if it has been defined.

**.ERROR**

**make** executes the recipe associated with this target whenever it detects an error condition.

**.EXPORT**

All prerequisites associated with this target which correspond to macro names are exported to the environment at the point in the makefile at which this target appears.

**.GROUPEPILOG**

**make** adds the recipe associated with this target after any group recipe for a target that has the **.EPILOG** attribute.

**.GROUPPROLOG**

**make** adds the recipe associated with this target before any group recipe for a target that has the **.PROLOG** attribute.

**.IMPORT**

**make** searches in the environment for prerequisite names specified for this target and defines them as macros with their value taken from the environment. If the prerequisite **.EVERYTHING** is given, **make** reads in the entire environment (see **-e** and **-E** options).

**.INCLUDE**

Parses another makefile just as if it had been located at the point of the **.INCLUDE** in the current makefile. The list of prerequisites gives the list of makefiles to read.

**.INCLUDEDIRS**

The list of prerequisites specified for this target defines the set of directories to search when including a makefile.

**.MAKEFILES**

The list of prerequisites is the set of files to try to read as the user makefile. These files are made in the order they are specified (from left to right) until one is found to be up to date. This is the file that is used.

**.POSIX**

This target may have no prerequisites and no recipes associated with it. Process the makefile as specified in the POSIX.2 draft standard. This special target must appear before the first non-comment line in the makefile. The target does the following:

- causes **make** to use the shell when executing *all* recipe lines; **make** invokes one shell per line, regardless of the setting of *SHELLMETAS*.
- disables brace expansion (set with the **.BRACEEXPAND** special target).
- disables meta-rule inferencing.
- disables conditionals.
- disables dynamic prerequisites.
- disables group recipes.
- **make** does *not* check for the string  $\$(MAKE)$  when run with the **-n** options specified.

**.REMOVE**

**make** uses the recipe of this target to remove any intermediate files that it creates if an error is encountered before creating the final target. This does not remove files marked **.PRECIOUS** or files that existed before **make** began execution.

**.SOURCE**

The prerequisite list of this target defines a set of directories to check when trying to locate a target file name.

**.SOURCE .x**

Same as **.SOURCE**, except that **make** searches the **.SOURCE .x** list first when trying to locate a file matching a target with a name that ends in the suffix **.x**.

**.SUFFIXES**

The prerequisite list of this target defines a set of suffixes to use when trying to infer a prerequisite for making a target.

A name of the form *library(member)* indicates a member of a library. The *library* portion is a target with the **.LIBRARY** attribute and the *member* portion is a prerequisite of the library target.

A name of the form *library((entry))* indicates the library module that contains the given entry point. Once again, the library portion is a target with the **.LIBRARY** attribute. **make** regards the library member that contains the entry point *entry* as a prerequisite of the library target.

**Control Macros**

**make** defines a number of control macros that alter its behavior. When there are several ways of doing the same thing, control macros are usually the best. A control macro that has the same function as a special target or attribute also has the same name.

Macros which are said to be *defined internally* are automatically created by **make** and you can use them with the usual  $\$(name)$  construct. For example, you can use  $\$(PWD)$  to obtain the current directory name.

Recognized control macros are:

**DIRSEPSTR**

Contains the characters used to separate parts in a path name and can be set by the user. **make** uses the first character in this string to build path names when necessary.

**.EPILOG**

If assigned a non-null value, the **.EPILOG** attribute is given to every target.

**GROUPFLAGS**

Specifies options to pass to **GROUPSHELL** when **make** invokes it to execute a group recipe.

**GROUPSHELL**

Gives the path name of the command interpreter (shell) that **make** calls to process group recipes.

**GROUPSUFFIX**

Specifies a string for **make** to use as a suffix when creating group recipe files to be handed to the command interpreter.

**.IGNORE**

If this is assigned a non-null value, **make** assigns the **.IGNORE** attribute to every target.

**INCDEPTH**

The current depth of makefile inclusion. This is set internally.

**MAKE** This is set by the startup file and may be changed by the user. The standard startup file defines it as

```
$(MAKECMD) $(MFLAGS)
```

The **MAKE** macro is not used by **make** itself, but the string **\$(MAKE)** is recognized when using the **-n** option for single line recipes.

**MAKECMD**

The name with which **make** was invoked.

**MAKEDIR**

Full path name of the initial directory in which **make** began execution.

**MAKEFLAGS**

The **MAKEFLAGS** macro contains all the options and macros specified in the **MAKEFLAGS** environment variable plus all the options and macros specified on the command line, with the following exceptions:

- Specifying **-c**, **-f**, or **-p** in the environment variable results in an error.
- These same options, when specified on the command line, do not appear in the **MAKEFLAGS** macro.

Options in the **MAKEFLAGS** environment variable may have leading minus signs and can be separated by spaces. These are stripped out when the **MAKEFLAGS** macro is constructed.

**Note:** **make** always reads the **MAKEFLAGS** environment variable before reading the makefile. The **-E** and **-e** options do not affect this.

**MAKESTARTUP**

Has the default value

`/etc/startup.mk`

To change where **make** looks for its startup file, you can set the environment variable *MAKESTARTUP* before running **make**. Since **make** processes command line macros after reading the startup file, setting this macro on the command line does not have the desired effect.

**MFLAGS**

Same as MAKEFLAGS, except that it includes the leading switch character.

**NULL** Permanently defined to be the NULL string.

**.PRECIOUS**

If this is assigned a non-null value, **make** assigns the *.PRECIOUS* attribute to every target.

**.PROLOG**

If this is assigned a non-null value, **make** assigns the *.PROLOG* attribute to every target.

**PWD** Full path name of the current directory in which **make** is executing.

**SHELL** Specifies the full path name of the command interpreter that **make** calls to process single line recipes, when necessary. **make** passes recipe lines to this shell only if they contain one or more of the characters given in *SHELLMETAS*; otherwise, it executes them directly. By default, the value of the *SHELL* environment variable does not affect the value of this macro; however, you can use the *.IMPORT* special target to assign the environment variable's value to this macro. You can also use the *EXPORT* special target to assign this macro's value to the *SHELL* environment variable.

**SHELLFLAGS**

Specifies options to pass to the shell when invoking it to execute a single line recipe.

**SHELLMETAS**

Specifies a list of metacharacters that can appear in single recipe lines. If **make** finds any metacharacter, it invokes the recipe using the shell specified by *SHELL*; otherwise, it executes the recipe without the shell.

**.SILENT**

If this is assigned a non-null value, **make** assigns the *.SILENT* attribute to every target.

## Making Libraries

A library is a file containing a collection of object files. To make a library, you specify it as a target with the `.LIBRARY` attribute and list its prerequisites. The prerequisites should be the object members that are to go into the library. When **make** makes the library target, it assigns the `.LIBRARYM` attribute to the prerequisites. This tells the file search mechanism to look for the member in the library if it cannot find an appropriate object file.

**make** tries to handle the old library construct format in a sensible way. When it finds *lib(member)*, it declares the *lib* portion as a target with the `.LIBRARY` attribute and the *member* portion as a prerequisite of the *lib* target. To make the library properly, old makefile scripts using this format must name the *lib* as a target and must try to bring it up to date. The same thing happens for any target of the form *lib(entry)*. These targets have an additional feature in that the *entry* target has the `.SYMBOL` attribute set automatically.

## Conditionals

You specify the conditional expression as follows:

```
.IF expression
... if text ...
.ELSE
... else text ...
.END
```

or

```
.IF expression
... if text ...
.ELSIF expression2
... elsif text ...
.ELSE
... else text ...
.END
```

The `.ELSE` or `.ELSIF` portion is optional, and you can nest the conditionals (that is, the text may contain another conditional). The `.IF`, `.ELSE`, `.ELSIF`, and `.END` must start in the first column of the line. *expression* or *expression2* can have one of three forms:

```
string
```

is true if the given string is non-NULL,

```
string == string
```

is true if the two strings are equal, and



```
string != string
```

is true if the two strings are not equal. Typically, one or both strings contain macros, which **make** expands before making comparisons. **make** also discards white space at the start and end of the text portion before the comparison. This means that a macro which expands to nothing but white space is considered a NULL value for the purpose of the comparison. If a macro expression needs to be compared to a NULL string, compare it to the value of the macro \$(NULL). The text enclosed in the conditional construct must have the same format that it would have outside the conditional. In particular, **make** assumes that anything that starts with a tab inside the conditional is a recipe line. This means that you cannot use tabs to indent text inside the conditional (except, of course, for recipe lines which always begin with tabs).

## ENVIRONMENT VARIABLES

**make** uses the following environment variable:

### *MAKEFLAGS*

contains a series of **make** options which are used as the default options for any **make** command. You may specify the options with or without leading minus signs (-) and blanks between them. It may also include macro definitions of the form usually found on the command line.

### *MAKESTARTUP*

contains the path name of the **make** startup file. By default, **make** uses the file `/etc/startup.mk` as its startup file. To use a different file, set this environment variable before running **make**.

*SHELL* contains a name of a command interpreter. To assign this value to the control macro SHELL, use the `.IMPORT` special target. You can also use the `.EXPORT` special target to assign the value of the SHELL macro to this environment variable.

## FILES

**make** uses the following file:

```
/etc/startup.mk
 default startup file containing default rules.
```

## DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- 1 Returned if you specified `-q` and file is not up to date.
- 255 An error occurred.

---

## Messages

- Message:** .ELSE without .IF  
**Cause:** You specified a .ELSE statement without a corresponding .IF statement.  
**Action:** Provide the corresponding .IF and .END statements (if necessary), or remove the .ELSE statement.
- Message:** .IF .ELSE ... .END nesting too deep  
**Cause:** The nesting of .IF .ELSE ... .END structures is too deep.  
**Action:** Modify your makefile so that these structures are not nested as deep.
- Message:** <+ diversion cannot be nested  
**Cause:** You attempted to put one <+ diversion inside another <+ diversion. **make** does not permit this.  
**Action:** Remove the nested <+ diversion.
- Message:** <+ diversion unterminated  
**Cause:** You specified a <+ to begin a diversion, but did not specify the corresponding +> to end it.  
**Action:** Provide the closing +>.
- Message:** <+ missing before +>  
**Cause:** You specified a +> to end a diversion before specifying the corresponding <+ to begin it.  
**Action:** Ensure that corresponding <+ and +> symbols appear in the correct order. Provide missing <+ or +> symbols, if necessary.
- Message:** Ambiguity in % targets [*targets*], chose target *target*  
**Cause:** There was more than one metarule that matched the target. The error message shows the rule that **make** will actually use.  
**Action:** Remove the extraneous metarules from the makefile.
- Message:** *attribute* ignored on special target [*special\_target*]  
**Cause:** You tried to specify attributes for a special target which does not take attributes.  
**Action:** Remove the special target or attribute from the offending rule.
- Message:** Attributes possibly ignored  
**Cause:** A special target may inherit attributes, but only certain attributes take effect on specific special targets.  
**Action:** Refer to *DESCRIPTION* section for more information about which attributes may be applied to which special targets.
- Message:** Can't touch library member  
**Cause:** **make** failed when trying to update the time stamp of an archive library member.  
**Action:** Ensure that the target file is a valid archive and that you have permissions to update it.

- 
- Message:** Cannot find member defining *archive((symbol))*  
**Cause:** **make** tried to locate the library archive member containing the symbol *symbol*.  
**Action:** Check that you archive file contains all the expected member files.
- Message:** Cannot mix single and group recipe lines  
**Cause:** You attempted to mix recipe lines with group recipes for the same rule.  
**Action:** Either make the entire recipe a group, or remove the group.
- Message:** Cannot open '*pathname*'  
**Cause:** **make** was unable to open a temporary file for a diversion or group recipe. You may not be able to write to your *TMPDIR* directory.  
**Action:** Make sure that the *TMPDIR* environment variable is set up properly, that you have the appropriate permissions in that directory and that there is space on the file system.
- Message:** Cannot use *-c*, *-f*, or *-p* options in MAKEFLAGS.  
**Cause:** You specified the *-c*, *-f*, or *-p* option in the environment variable *MAKEFLAGS*.  
**Action:** Do not specify these options in *MAKEFLAGS*.
- Message:** Configuration file '*filename*' not found  
**Cause:** **make** was unable to find the configuration file *filename*.  
**Action:** Check that the configuration file exists, was named properly and that you have the appropriate permissions. Also check the value of the variable *MAKESTARTUP*.
- Message:** Could not export *macro*  
**Cause:** **make** was unable to export the name macro to the environment.  
**Action:** There may not be sufficient system resources to do the export. Free up some resources and try again.
- Message:** Don't know how to make *target*  
**Cause:** **make** was unable to find or infer a recipe to build the specified target.  
**Action:** Make sure that there is a rule for this target in your makefile.
- Message:** Duplicate entry [*prerequisite*] in prerequisite list  
**Cause:** You have specified a prerequisite for a target more than once.  
**Action:** Remove the duplicate prerequisite.
- Message:** Duplicate entry [*target*] in target list  
**Cause:** You specified a target in a target list more than once.  
**Action:** Remove the duplicate entry.
- Message:** Empty recipe for special target *special\_target*  
**Cause:** The special target specified requires that a recipe also be specified for it.  
**Action:** Refer to the documentation for the target, and add an appropriate recipe.

- 
- Message:** file "*filename*": system error  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** File *filename* not found  
**Cause:** **make** was unable to find the file *filename*.  
**Action:** Check that the file exists, was named properly and that you have the appropriate permissions.
- Message:** Found unmatched ' ]'  
**Cause:** You specified a ] in your makefile for a group recipe without providing the matching [.  
**Action:** Provide the missing [.
- Message:** GROUPSHELL macro not defined  
**Cause:** You attempted to execute a recipe that required the shell and the GROUPSHELL macro was not defined.  
**Action:** Make sure that the GROUPSHELL macro is defined properly in your makefile or startup.mk file.
- Message:** Imported macro '*macro*' not found in environment  
**Cause:** **make** attempted to import a macro that was not present in the program environment.  
**Action:** Define the appropriate environment variable, remove the import rule or add the .IGNORE attribute to the import rule.
- Message:** Include file *filename*, not found  
**Cause:** **make** was unable to find the file *filename*.  
**Action:** Check that the file exists, was named properly and that you have the appropriate permissions. Also check the prerequisites of the .INCLUDIRS target to make sure that it specifies the correct path.
- Message:** Incomplete rule recipe group detected  
**Cause:** You specified a group recipe but omitted the closing ].  
**Action:** Add the closing square bracket.
- Message:** Inference rules result in circular dependency for [*target*]  
**Cause:** When performing transitive closure on a set of metarules, **make** discovered that *target* has itself as a dependent (that is, a circular dependency).  
**Action:** Remove or redefine one of the inference rules causing the circular dependency.

- 
- Message:** Invalid library format  
**Cause:** **make** attempted to access a library that was not in the correct format.  
**Action:** Verify that your library is correct and rebuild it if necessary.
- Message:** Line too long  
**Cause:** **make** encountered a line that was too long to process either when reading a file or performing macro expansions.  
**Action:** Restructure your makefile so that the line can be shortened.
- Message:** Missing argument for "-opt" option  
**Cause:** You specified the -opt option but did not follow it with the expected argument.  
**Action:** Provide the expected argument. Check the *Options* section of this man page for a list of valid options and their arguments.
- Message:** Missing .END for .IF  
**Cause:** You specified a .IF statement without the corresponding .END statement.  
**Action:** Provide the missing .END statement, or remove the extra .IF statement.
- Message:** Missing targets or attributes in rule  
**Cause:** **make** encountered a rule that had no targets or attributes specified when reading input.  
**Action:** Correct the syntax of your makefile.
- Message:** Multiple .SETDIR for *target* ignored  
**Cause:** More than one .SETDIR attribute was specified for the target. The subsequent .SETDIRs have been ignored.  
**Action:** Remove the extra .SETDIRs.
- Message:** Multiple targets are not allowed in % rules  
**Cause:** A metarule can have only one target specified.  
**Action:** Refer to the section in the manual on inference rules and correct the makefile.
- Message:** Multiply defined recipe for target *target*  
**Cause:** You specified more than one recipe for *target* in different rules, and the rules use the : operator.  
**Action:** Either use the ; operator to handle independent recipes, or correct your makefile.
- Message:** Name too long '*pathname*'  
**Cause:** You specified a path name which exceeds the maximum path name length for **make**. You can use .SETDIR and relative names to traverse the directory tree, as long as this limit is not exceeded.  
**Action:** Restructure your makefile.

---

**Message:** No .INCLUDE file(s) specified  
**Cause:** You specified a .INCLUDE special target without providing the names of the files to be included.  
**Action:** Refer to the documentation on the .INCLUDE target and add the missing file names.

**Message:** No file name for -f  
**Cause:** You specified the -f option but did not follow it with a file name.  
**Action:** Provide a file name following the -f option.

**Message:** No macro name  
**Cause:** A macro assignment = appears without a macro name.  
**Action:** Correct the offending line.

**Message:** No 'makefile' present  
**Cause:** **make** was unable to find Makefile or makefile, and did not have any default rules.  
**Action:** Create the missing makefile, or add default rules to startup.mk.

**Message:** No more memory  
**Cause:** **make** was unable to allocate storage space.  
**Action:** Free up some resources and try again.

**Message:** No target  
**Cause:** **make** had a makefile to process, but did not find a rule defining a target to be made.  
**Action:** Add a target rule to your makefile, or specify a target on the command line.

**Message:** No 'makefile' present  
**Cause:** **make** was unable to find Makefile or makefile, and did not have any default rules.  
**Action:** Create the missing makefile, or add default rules to startup.mk.

**Message:** Nonglobal attributes ignored  
**Cause:** You specified attributes that are nonglobal. **make** will ignore them.  
**Action:** Remove the offending attributes.

**Message:** Only a single % allowed in a target pattern  
**Cause:** A metarule target may contain only a single '%'.  
**Action:** Remove the additional percent signs.

**Message:** Only one .SETDIR attribute allowed in rule line  
**Cause:** You have a rule with more than one .SETDIR attribute.  
**Action:** If you want **make** to search for a file in a number of different directories, use the .SOURCE special target.

- 
- Message:** Openfile: bad name  
**Cause:** **make** attempted to open a file with an invalid or NULL name.  
**Action:** Edit the makefile and correct the file name.
- Message:** Operator after special target treated as ':'  
**Cause:** You specified a modifier, such as !, with a rule defining a special target. **make** ignores any such modifiers.  
**Action:** Remove the extraneous modifier.
- Message:** Option -c failed to change directory to "*pathname*": *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** Option *-option* argument missing  
**Cause:** You did not provide an argument for *-option*.  
**Action:** Provide the missing argument.
- Message:** reading file: *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** SHELL macro not defined  
**Cause:** You attempted to execute a recipe that required the shell and the SHELL macro was not defined.  
**Action:** Make sure that the SHELL macro is defined properly in your makefile or startup.mk file.
- Message:** Special target *target* cannot be a prerequisite  
**Cause:** You tried to use a special target as a prerequisite.  
**Action:** Edit the makefile, and remove the special target from the prerequisite list.
- Message:** Special target must appear alone  
**Cause:** Various special targets cannot appear with other targets in a rule. For example, a rule with .ERROR as a special target cannot mention any other target.  
**Action:** Correct the offending line.
- Message:** Syntax error in % rule, missing % target  
**Cause:** You specified your meta-rule incorrectly. The target must contain a %.  
**Action:** Correct the syntax of the rule.
- Message:** Target '*target*' cannot mix ':' and '::' rules  
**Cause:** You have defined a rule for target using the :: operator, and then followed this with another rule for target using the : operator.  
**Action:** Either modify the second rule to use :: or remove it.

- 
- Message:** Too many arguments -- limit *num*  
**Cause:** Too many arguments were produced when **make** tried to exec a line in a recipe.  
**Action:** Simplify the recipe line.
- Message:** Too many makefiles specified  
**Cause:** You specified too many files using the **-f** option.  
**Action:** Combine one or more files into a single file.
- Message:** Too many open files. Max nesting level is *num*.  
**Cause:** You have exceeded the maximum limit of `.INCLUDES`.  
**Action:** Check to see if you have recursively included a **make** file, or simplify your makefile.
- Message:** Unable to change directory to '*pathname*'  
**Cause:** **make** was unable to change directories to *pathname*.  
**Action:** Check that the directory name specification is correct and that you have the appropriate permissions for the directory.
- Message:** Unable to determine current directory  
**Cause:** **make** was unable to find out what its current directory was.  
**Action:** Verify that you have all necessary permissions to determine your current directory.
- Message:** Unable to return to directory '*pathname*'  
**Cause:** **make** was unable to change directories.  
**Action:** Check directory path specification and verify that you have the required permissions.
- Message:** Unknown option "*-option*"  
**Cause:** You specified an option that is not valid for this command.  
**Action:** Check the *DESCRIPTIONS* section for a list of valid options.
- Message:** Unmatched `.END`  
**Cause:** You specified a `.END` statement without the corresponding `.IF` statement.  
**Action:** Provide the missing `.IF` statement, or remove the extra `.ELSE` statement.
- Message:** Unmatched "quote  
**Cause:** You specified an opening " on a line that did not contain a closing ".  
**Action:** Correct the offending line.
- Message:** Write error on temp file  
**Cause:** An error occurred while trying to write on a diversion or group recipe temporary file.  
**Action:** Ensure that there is space on the file system containing the temporary file.



**PORTABILITY**

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

The following features of MAKE are enhancements to POSIX.2:

- The following options; **-c** *dir*, **-E**, **-u**, **-V**, **-v**, **-x**.
- The **-n** option has enhanced functionality not covered by the standard; for more information see the explanations of the **-n** option and the .POSIX special target in this man page.
- The following run-time macros;  $\$&$ ,  $\$^$ ,  $\$>$ .
- The following dynamic prerequisites;  $\$ \$%$ ,  $\$ \$>$ ,  $\$ \$*$ ,  $\$ \$@$ .
- All macro expansions.
- Macro assignments of the following forms:

```
macroname := stringassigned
macroname += stringassigned
```

- Brace expansion.
- Backslash continuation.
- The quoting mechanism, as in the following example:

```
"a:target" : "a:prerequisite"
```

- All rule operators except the colon (:).
- Conditionals.
- Meta-Rules.
- All MAKE attributes *except*; .IGNORE, .PRECIOUS, .SILENT (referred to in POSIX.2 as special targets).
- All MAKE special targets *except*; .DEFAULT, .POSIX, .SUFFIXES (referred to in POSIX.2 as special targets).
- All MAKE control macros *except* SHELL (referred to in POSIX.2 as control macros).

**LIMITS**

No single makefile script line can be longer than 8192 chars. In some environments the length of an argument string is restricted.

**MPE/iX NOTES**

The current MPE/iX implementation of **make** does not understand the format of libraries. As a result, it is unable to extract the date/time stamp for an archive and thus, cannot compare that date/time stamp to those of other files.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

**NAME**

man — print sections of the online reference manual

**SYNOPSIS**

**man** [-wx] [-M path] [section] entry ...

**man** [-kwx] [-M path] keyword ...

**DESCRIPTION**

The **man** command either prints portions of the online manual to the standard output or searches for manual entries having the specified keywords associated with them.

**man** recognizes the following options:

**-k** searches a precomputed database of synopsis lines for information on *keywords*.

**-M path**

searches the directories indicated by *path* for manual entries. If **-M** is not specified, **man** uses the path specified in the *MANPATH* environment variable if set; otherwise **man** searches */usr/man*. All manual entries are found by searching similarly structured file trees rooted at one or more places. See the *FILES* section for a description of what files and directories **man** should find in each directory that it searches.

**-w** only displays the file name of the file containing the manual entry.

**-x** displays what files **man** is searching to find the manual entry.

*section* is a number (0-9) representing a section of the manual. **man** will search within the specified section for *entry*.

Normally, **man** displays each specified *entry* of the manual. You can instruct **man** to only search a given section of the manual by specifying the section number on the command line. The online reference manual for MPE/iX Shell and Utilities contains five basic sections:

1. Commands
2. System Interface Calls
3. C Runtime Library
4. File Formats
5. Miscellaneous Information

To find a given entry, **man** follows these search rules. If you specified a *section*, **man** searches for the appropriate entry in that section of the manual; otherwise **man** looks for the first entry named *entry* regardless of the section. If any rule results in finding the man page, **man** displays the entry and exits.

- **man** checks each directory in *MANPATH* for a file named `man.dbz`. If it exists, **man** looks for the requested entry in its index (see *File Format*).
- For each possible section (that is, *section* if you specified it, or all sections in order from 1 through 9, then 0 if you did not):
  - **man** checks each directory in *MANPATH* for a file named `catn/entry.n`, where *n* is the section number. If it exists, **man** checks to see if it was compressed with **pack** or **compress**, and uncompresses it (calling **pcat** if the file was **packed**.)
  - **man** checks each directory in *MANPATH* for a file named `mann/entry.n`. This is the unformatted manual entry. It is assumed to be in **troff** format.

If output is to the terminal, then **man** invokes a pager command to filter and display the manual pages. If *MANPAGER* is defined, it is used. If not, then if *PAGER* is defined, it is used; otherwise, **man** defaults to using the command:

```
more -A -s
```

### File Format

The manual files are normally kept in a single large file, called `man.dbz`. The file starts with a magic text string:

```
!<man database compressed>\n
```

and continues with the index:

```
14 bytes formatted man page name
9 bytes seek pointer
9 bytes length
```

The name is simply the entry name, followed by dot and the section number, for example, this man page would be named **man.1**. When **man** finds a matching entry, it then seeks to the point in the file specified by the given seek pointer, and uncompresses for length bytes. Each manual entry is compressed separately.

### EXAMPLES

In order to find out which utilities do comparisons, it might be useful to type:

```
man -k compare
```

## ENVIRONMENT VARIABLES

The following environment variables affect **man**:

*MANPATH*

contains a list of paths to search for man pages.

*MANPAGER*

*PAGER* contains an output filtering command for use when displaying manual entries on a terminal.

*TMPDIR*

identifies the directory where temporary files reside.

## FILES

**man** uses the following files and directories:

`/usr/man`

default directory for online manual

`cat[0-9]/*. [0-9]`

subdirectories containing pre-formatted manual entries in normal, compressed, or packed form.

`man[0-9]/*. [0-9]`

unformatted manual entries.

`whatis`

database used by **-k** option.

`man.dbz`

master file containing all manual entries.

## DIAGNOSTICS

Possible exit status values are:

0 Successful completion.

1 An error occurred.

## Messages

**Message:** `child process creation: system error`

**Cause:** See **syserror(3)**.

**Action:** See **syserror(3)**.

**Message:** `execute "filename": system error`

**Cause:** See **syserror(3)**.

**Action:** See **syserror(3)**.

- 
- Message:** insufficient memory: *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** *keyword*: nothing appropriate  
**Cause:** **man** found no entries for the specified *keyword* in the `what is` database.  
**Action:** Check that you spelled *keyword* correctly. Try a related *keyword*.
- Message:** Missing path after `-M`  
**Cause:** You specified the `-M` option but did not provide a path as an argument.  
**Action:** Provide the missing path.
- Message:** No entry for "*entry*" in section *sec* of the manual.  
**Cause:** **man** found no man page for *entry* in the specified section of the manual.  
**Action:** Check that you spelled *entry* correctly, that you specified the correct section of the manual, and that the **man** command will search the path containing its man page.
- Message:** No manual entry for "*entry*".  
**Cause:** **man** found no man page for *entry* in any section of the manual.  
**Action:** Check that you spelled *entry* correctly, and that **man** command searches the path containing its man page.
- Message:** no pipes available: *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** system call failed: *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** Unknown option "*-option*"  
**Cause:** You specified an option that is not valid for **man**.  
**Action:** Check the *DESCRIPTION* section for a list of valid **man** options.

Since **man** calls **uncompress** to decompress the `man.dbz` file, it is also possible to receive an **uncompress** error. See the **uncompress**(1) man page for more information.

## PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

The elements of the environment variable `MANPATH` are separated by colons.

The `-M` option, the `-x` option, the `-w` option, the `MANPAGER` environment variable, the default pager, and the ability to specify *section* on the command line are all extensions to the POSIX standard.

**MPE/iX NOTES**

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

**SEE ALSO**

**compress(1), help(1), more(1), pack(1)**

**NAME**

merge — three-way file merge

**SYNOPSIS**

```
merge [-H] [-h] [-p] [-q] [-z] file1 root file2 [mark1 [mark2]]
```

**DESCRIPTION**

**merge** incorporates all changes that lead from *root* to *file2* into *file1*. The result is stored in *file1*.

**merge** is useful for combining separate changes to an original. Suppose *root* is the original, and both *file1* and *file2* are modifications of *root*. Then, **merge** combines both changes.

An overlap occurs if both *file1* and *file2* have changes in a common segment of lines. **merge** prints how many overlaps occurred, and includes both alternatives in the result. The alternatives are delimited as follows:

```
<<<<<<< file1
lines in file1
=====
lines in file2
>>>>>>> file2
```

If there are overlaps, you should edit the result and delete one of the alternatives.

Putting in a word or phrase for *mark1* and/or *mark2* lets you change what is printed to mark overlapping lines in *file1* and/or *file2*. You must use double quotes if you substitute in a phrase rather than a single word.

RCS users normally use **rcsmerge**(1) to access this functionality rather than calling **merge** directly.

**Options**

**merge** accepts the following options:

- H** tells **diff** to use the **-H** option when called by **merge**.
- h** tells **diff** to use the **-h** option when called by **merge**.
- p** sends the final result of the merge operation to the standard output. Normally, **merge** stores the result in *file1*.
- q** quiet mode; suppresses diagnostics.
- z** security mode; overwrites temporary files with null bytes before unlinking.

**DIAGNOSTICS**

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

**Messages**

- Message:** cannot find "diff3" command: *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** cannot rename "name1" to "name2": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** child process: *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** failure on dup() or dup2() call: *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** *filename*: *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** memory allocation failure in strdup(): *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** no temporary files available  
**Cause:** **merge** was unable to create a necessary temporary file, probably because it could not create a new unique name for the file.  
**Action:** Make sure that your system has a directory for storing temporary files (either /tmp, or another directory indicated by the variable *TMPDIR*), and that this directory has both space for new files and appropriate permissions.
- Message:** temporary file "*filename*": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.
- Message:** temporary output file: *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.



**Message:** Unable to open file *filename* for zeroing: *system error*

**Cause:** See **syserror**(3).

**Action:** See **syserror**(3).

**Message:** Unable to stat file *filename* when unlinking: *system error*

**Cause:** See **syserror**(3).

**Action:** See **syserror**(3).

**Message:** Unknown option "*-option*"

**Cause:** You specified an option that is not valid for **merge**.

**Action:** Check the *DESCRIPTION* section for a list of valid **merge** options.

**Message:** Write error when zeroing file *filename*: *system error*

**Cause:** See **syserror**(3).

**Action:** See **syserror**(3).

#### PORTABILITY

All UNIX systems.

The options **-H** and **-h** are extensions to traditional implementations of **merge**. In traditional RCS implementations, **merge** is a shell script.

#### LIMITS

Limits are those of **diff3**(1).

#### MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

#### SEE ALSO

**co**(1), **diff**(1), **diff3**(1), **rcsmerge**(1)

**NAME**

mesg — allow or refuse messages

**SYNOPSIS**

**mesg** [y] [n]

**DESCRIPTION**

**Note:** The MPE/iX implementation of this utility does not function exactly as this man page describes. For details, see the *MPE/iX NOTES* section at the end of this man page.

**mesg** determines whether other users can send messages to your terminal with **talk**, **write**, or similar utilities.

mesg y

lets other people send you messages.

mesg n

tells the system not to let others send you messages.

mesg

outputs the current setting without changing it.

The terminal is determined by the first of standard input, standard output, or standard error which is directed to a terminal.

**DIAGNOSTICS**

Possible exit status values are:

- 0 Receiving messages is currently allowed.
- 1 Receiving messages is not currently allowed.
- 2 An error occurred.

**Messages**

**Message:** Not setgid to *term\_group* -- contact your system administrator

**Cause:** The process does not belong to the *term\_group* group.

**Action:** Contact your system administrator.

- 
- Message:** *term\_group* group is missing -- contact your system administrator
- Cause:** **mesg** was unable to find *term\_group* (that is, the group that owns all the terminals).
- Action:** Contact your system administrator.
- Message:** terminal "*term*": *system error*
- Cause:** See **syserror**(3).
- Action:** See **syserror**(3).
- Message:** Unknown operand "*string*"
- Cause:** You specified an operand *string* that was not the required yes or no expression (normally y or n but may differ for other languages).
- Action:** Specify y or n as operands to **mesg**.
- Message:** unknown terminal
- Cause:** **mesg** was unable to determine a path for the terminal name, or none of standard input, standard output, and standard error were a terminal.
- Action:** Make sure that standard input is a terminal, not a redirected file or a pipe.

**PORTABILITY**

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

**MPE/iX NOTES**

The **mesg** utility is currently implemented on MPE/iX as a call to the MPE/iX **SETMSG** command. As a result, it has the same behavior as the **SETMSG** command. For more information about the MPE/iX CI **SETMSG** command, refer to the *MPE/iX Commands Reference Manual*.

Executing

```
mesg n
```

disables messages from **write**, **wall**, and the MPE/iX CI **TELL** command. It uses the MPE/iX CI variable HPQUIET to determine the current status.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

**SEE ALSO**

**talk**(1), **write**(1)

**NAME**

mkdir — create a new directory

**SYNOPSIS**

**mkdir** [-p] [-m *mode*] *directory*...

**DESCRIPTION**

The **mkdir** command creates a new directory for each named *directory* argument.

**Option**

**mkdir** accepts the following options:

**-m** *mode*

lets you specify permissions for the directories. The *mode* argument may have the same value as the *mode* for **chmod**; see **chmod(1)** for more details.

**-p**

creates intermediate directory components that don't already exist. For example, if one of the *directory* arguments is *dir/subdir/subsub* and *subdir* doesn't already exist, **mkdir** creates it. Directories are created with the *mode* *u+wx*, which means read, write, and search permissions for the owner.

**DIAGNOSTICS**

Possible exit status values are:

0 Successful completion.

1 An error occurred.

**Messages**

**Message:** directory "*pathname*": *system error*

**Cause:** See **syserror(3)**.

**Action:** See **syserror(3)**.

**Message:** Insufficient memory

**Cause:** There were not enough free system resources to perform the specified operation.

**Action:** Free up more resources.

**Message:** Missing mode after -m

**Cause:** You specified the **-m** option but did not follow it with a mode.

**Action:** Provide a valid mode following the **-m** option.

- 
- Message:** Octal mode may contain only digits [0-7] in *numstring*
- Cause:** When using the octal mode to indicate new access permissions, you specified a string *numstring* which contained a character other than the digits 0 to 7.
- Action:** Make sure that all *mode* values in octal mode are valid octal numbers, containing only the digits 0 through 7.
- Message:** Unknown option "*-option*"
- Cause:** You specified an option that is not valid for **mkdir**.
- Action:** Check the *DESCRIPTION* section for a list of valid **mkdir** options.
- Message:** Unknown or missing operator in symbolic mode "*modestring*" "
- Cause:** When using the symbolic mode to indicate new access permissions, you specified a string *modestring* which had either a missing or unrecognized operator.
- Action:** Make sure that all *mode* values in symbolic mode contain one of the following operators: +, -, or =.

#### PORTABILITY

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

#### MPE/iX NOTES

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

#### SEE ALSO

**rm(1)**, **rmdir(1)**, **umask(1)**

**NAME**

mkfifo — create a FIFO special file

**SYNOPSIS**

**mkfifo** [-p] [-m *mode*] *file* ...

**DESCRIPTION**

**mkfifo** creates one or more FIFO special file with the given names.

**Options**

**mkfifo** accepts the following options:

**-m** *mode*

lets you specify file permissions for the files. The *mode* argument may have the same value as the *mode* for **chmod**; see **chmod**(1) for more details.

**-p**

creates intermediate directory components that don't already exist. For example, if one of the *file* arguments is *dir/subdir/file* and *subdir* doesn't exist already, this option creates it. Directories are created with the *mode* *u+wx*, which means read, write, and search permissions to the owner.

**DIAGNOSTICS**

Possible exit status values are:

0 Successful completion.

1 An error occurred.

**Messages**

**Message:** fifo file "*filename*": system error

**Cause:** See **syserror**(3).

**Action:** See **syserror**(3).

**Message:** Insufficient memory

**Cause:** There were not enough free system resources to perform the specified operation.

**Action:** Free up more resources.

**Message:** Missing mode after -m

**Cause:** You specified the **-m** option without providing the mode argument.

**Action:** Provide the missing mode.

- 
- Message:** Octal mode may contain only digits [0-7] in *numstring*
- Cause:** When using the octal mode to indicate new access permissions, you specified a string *numstring* which contained a character other than the digits 0 to 7.
- Action:** Make sure that all *mode* values in octal mode are valid octal numbers, containing only the digits 0 through 7.
- Message:** Unknown option "*-option*"
- Cause:** You specified an option that is not valid for **mkfifo**.
- Action:** Check the *DESCRIPTION* section of this man page for a list of valid **mkfifo** options.
- Message:** Unknown or missing operator in symbolic mode "*modestring*"
- Cause:** When using the symbolic mode to indicate new access permissions, you specified a string *modestring* which had either a missing or unrecognized operator.
- Action:** Make sure that all *mode* values in symbolic mode contain one of the following operators: +, -, or =.

**PORTABILITY**

POSIX.2.x/OPEN Portability Guide 4.0. All UNIX systems.

The **-p** option is an extension to the POSIX standard.

**MPE/iX NOTES**

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

**SEE ALSO**

**chmod(1)**, **create(1)**, **mkdir(1)**, **mknod(1)**, **umask(1)**

**NAME**

mknod — build a special file

**SYNOPSIS**

**mknod** *pathname* **b** *major* *minor*

**mknod** *pathname* **c** *major* *minor*

**mknod** *pathname* **p**

**DESCRIPTION**

**mknod** creates a *special* file with the given *pathname*. **b** indicates block-special files (for example, disks and tapes), while **c** indicates character-special files (for example, printers and other devices). *major* gives the major device type and *minor* the minor device type; device types can be either octal or decimal numbers. Device type must be obtained from the system source file `conf.c`.

The final form of **mknod** in the *SYNOPSIS* section (**p**) creates a FIFO special file (that is, a named pipe).

**DIAGNOSTICS**

Possible exit status values are:

- 0 Successful completion.
- 1 An error occurred.

**Messages**

**Message:** block special file "*filename*": *system error*

**Cause:** See **syserror(3)**.

**Action:** See **syserror(3)**.

**Message:** character special file "*filename*": *system error*

**Cause:** See **syserror(3)**.

**Action:** See **syserror(3)**.

**Message:** fifo file "*filename*": *system error*

**Cause:** See **syserror(3)**.

**Action:** See **syserror(3)**.

**Message:** Missing major/minor device

**Cause:** You failed to specify the major or minor device type argument for a character or block special file.

**Action:** Provide the missing argument.



**PORTABILITY**

All UNIX systems. Within POSIX, **mknod** has been superseded by **mkfifo** for pipes. The POSIX family of standards have not yet designed an alternative to **mknod** for special files.

**MPE/iX NOTES**

For information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

**SEE ALSO**

**mkfifo(1)**

**NAME**

more — display files on a page-by-page basis

**SYNOPSIS**

```
more [-ceiSS] [-A] [-u] [-n number] [-P prompt] [-p command] [-t tag] [file ...]
more [-ceiSS] [-A] [-u] [-n number] [-P prompt] [-t tag] [+command] [file ...]
```

**DESCRIPTION**

**more** displays files one page at a time. It obtains the number of lines per page from the environment or from the **-n** option. If the standard output is not a terminal device, the number of lines per page is infinite.

**more** displays the files specified by *file ...* (that is, a list of file names) one at a time. When **more** finishes displaying one file, it begins displaying the next one in the list. If you give **-** as one of the file names, **more** reads the standard input at that point in the sequence.

**more** allows paging forwards and backwards (if possible) and searching for strings.

**Options**

**more** accepts the following options:

- A** causes the display of all characters, including unprintable ones. Normally unprintable characters are displayed in a printable format. Further, ANSI escape sequences for display modes are processed. This option cannot be used with **-u**.
- c** clears the screen before displaying a new file. If at any time, the new screen to be displayed does not have any lines in common with the current screen, **more** does not scroll, but instead, redraws the screen one line at a time, starting from the top. **more** may ignore this option if the terminal doesn't support such operations.
- e** exits immediately after displaying the last line of the last file. Normally, if standard output is a terminal device, **more** stops after displaying the last line of the last file and prompts for a new command. If the command that displays text causes **more** to reach the end of the file again, **more** exits.
- i** ignores case during searches.
- n number**  
specifies the number of lines per page. This overrides any values obtained from the environment.
- P string**  
sets the prompt that appears at end of each page of text to *string*. The default prompt is [*filename*]. **more** normally displays the prompt in STANDOUT mode.

**-p** *command*

**+command**

initially executes the **more** command on each file. If it executes successfully and *command* is a positioning command such as a line number or a regular expression search, **more** displays the resulting page; otherwise **more** displays the first page of the file. If both the **-t** and **-p** options are specified, the **-t** option is processed first.

**-s** displays the prompt in normal mode rather than STANDOUT mode.

**-s** replaces consecutive empty lines with a single empty line.

**-t tag** searches for the named *tag* and displays the page of text containing it. See **ctags(1)** for more information.

**-u** displays all backspaces as ^H. Normally *characterbackspace\_*(underscore) displays *character* as underlined and *characterbackspacecharacter* displays *character* as boldfaced. **-u** also displays all carriage returns as ^M. This option cannot be used with **-A**.

### Interactive Commands

**more** also accepts the following interactive commands.

[*n*]**b**

[*n*]**CTRL-B**

[*n*]**PgUp**

moves backward *n* lines, with a default of one page. If *n* is more than the page size, **more** displays only the final page.

[*n*]**d**

[*n*]**CTRL-D**

scrolls forward *n* lines, with a default of one half of the page size. If you specify *n*, it becomes the new default for subsequent **d** and **u** commands.

[*n*]**f**

[*n*]**CTRL-F**

[*n*]**PgDn**

moves forward *n* lines, with a default of one page. At end-of-file, **more** continues with the next file in list, or exits if the current file is the last one in the list.

[*n*]**G**

goes to the *n*th line in the file. If you do not specify *n*, **more** advances to the end of the file.

[*n*]**g**

goes to the *n*th line in the file, with the default being the first line of the file.

**h**

displays a summary of interactive commands.

- [*n*]**j**  
[*n*]**SPACE**  
[*n*]**ENTER**  
[*n*]↓ scrolls forward *n* lines, with a default of one line for **j**, **ENTER**, and ↓ and a default of one page for **SPACE**. This command displays the entire *n* lines even if *n* is more than the page size. At end-of-file, these commands cause **more** to begin displaying the next file in the list, or to exit if the current file is the last one in the list.
- [*n*]**k**  
[*n*]↑ scrolls backward *n* lines, with a default of one line. This command displays the entire *n* lines even if *n* is more than the page size.
- mletter** marks the current position with the lowercase *letter*. When you view a new file, all previous marks are lost.
- [*n*]**N** repeats the previous search, but in the opposite direction. If you specify *n*, **more** repeats the search *n* times.
- [*n*]**n** repeats the previous search. If you specify *n*, **more** repeats the search *n* times. For example if there are eight occurrences of *pattern* in the file and */pattern* found the second occurrence then a follow-up command of **5n** finds and sets the current position to the 7th occurrence of *pattern*.
- q**  
**:q**  
**ZZ** exits **more**.
- R** refreshes the screen and discards any buffered input.
- r**  
**CTRL-L** refreshes the screen.
- [*n*]**s** skips forward *n* lines (with a default of one line) and displays one page beginning at that point. If *n* would cause less than a full page to be displayed, **more** displays the last page in the file.
- [*n*]**u**  
[*n*]**CTRL-U** scrolls backward *n* lines, with a default of one half of the page size. If you specify *n*, it becomes the new default for subsequent **d** and **u** commands.
- v** invokes an editor to edit the current file. **more** uses the editor named by the environment variable *EDITOR*. The default editor is **vi**.
- 'letter* returns to the position marked with *letter*.

” returns to the position from which you last issued a movement command of greater than one page or the beginning of the file if you have issued no such commands.

**:e** *[filename]*ENTER

stops viewing the current file and views *filename* instead. If you do not specify *filename*, **more** returns to the beginning of the current file. If *filename* is #, **more** returns to the last file viewed before the current one.

**[n]:n** views the next file from the list given on the command line. If you specify *n*, **more** views the *n*th next file from the list.

**[n]:p** views the previous file from the list given in the command line. If you specify *n*, **more** views the *n*th previous file from the list.

**:t** *tagname*

goes to *tagname* (see **ctags**(1)).

**:w** *filename*

writes the contents of the current file to the file *filename*.

**!<shell command>**

escapes to shell and executes *shell command*.

=

**CTRL-G**

displays, where possible, the name of the file currently being viewed, its number (relative to the total number of files specified in the command line), the current line number, the current byte number, the total bytes to display and what percentage of the file has been displayed.

**[n]/[!]pattern**

searches forward in the file for the *n*th line containing *pattern*. *n* defaults to one if not specified. If *pattern* is the null regular expression (*/*), **more** uses the previous *pattern*. If the character **!** precedes *pattern*, **more** searches for lines that do not contain *pattern*.

**[n]?[!]pattern**

searches backward in the file for the *n*th line containing *pattern*. The search begins at the line immediately before the top line displayed. *n* defaults to one if not specified. If *pattern* is the null regular expression (*?*), **more** uses the previous *pattern*. If the character **!** precedes *pattern*, **more** searches for lines that do not contain *pattern*.

**HOME** goes to the first line in the file.

**END** goes to the last line in the file.

---

## ENVIRONMENT VARIABLES

The following environment variables affect the operation of **more**:

### *COLUMNS*

contains the maximum number of columns to display on one line.

### *EDITOR*

contains the name of the editor that the **v** command invokes.

*LINES* contains the number of lines in a page. This value takes precedence over value from *TERM*; however, the **-n** value takes precedence over the *LINES* value.

### *MORE*

contains a list of options (from those listed in the *DESCRIPTION* section) as they would appear on the command line. This variable takes preference over the *TERM* and *LINES* variables.

### *TERM*

contains the name of the terminal type.

## DIAGNOSTICS

Possible exit status values are:

- 0 Successful completion.
- >0 An error occurred.

## Messages

**Message:** `***** filename: Not a text file *****`

**Cause:** You specified the file *filename* which was not a text file.

**Action:** Specify a text file.

**Message:** Badly constructed regular expression.

**Cause:** **more** encountered a syntax in a regular expression.

**Action:** Check the syntax of the regular expression.

**Message:** Badly formed number in "*num*"

**Cause:** You specified an option which requires a numeric argument, but the argument given was not a valid number.

**Action:** Provide a valid number as an argument.

**Message:** cannot create file "*filename*"

**Cause:** You attempted to use the **:w filename** command, but **more** was unable to create the file, probably due to the existence of a file with that name, no space on the destination device, or inappropriate permissions on the destination directory.

**Action:** Check that *filename* does not already exist, that there is space on the destination device, and that you have appropriate permissions on the destination device.

- 
- Message:** Cannot edit standard input.  
**Cause:** You attempted to use the **v** command to edit text that was coming from the standard input stream.  
**Action:** Store the input text in a temporary file and then use **more** to view that file. This allows you to edit the text if necessary.
- Message:** cannot reopen input file  
**Cause:** You attempted to view a previously viewed file with the **:e** command, and when **more** tried to re-open that file, it could not, or you attempted to re-open the current file (due to a **R** command or the completion of a shell escape command) and it failed.  
**Action:** Find out what caused the file become unopenable, and fix the problem.
- Message:** input file "*filename*"  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** insufficient memory  
**Cause:** There were not enough free system resources for **more** to work properly.  
**Action:** Free up more system resources.
- Message:** invalid command  
**Cause:** The *command* argument to **-p** or **+** was invalid.  
**Action:** Fix the syntax of *command*.
- Message:** interactive terminal inaccessible  
**Cause:** **more** was unable to open the terminal for input.  
**Action:** Make sure that the standard input is assigned to a terminal.
- Message:** Mark must be a lowercase letter.  
**Cause:** You used the **m** command to mark a position in the file; however you attempted to name the mark with a character other than a lowercase letter.  
**Action:** Use lowercase letters for all mark names.
- Message:** Missing prompt after **-P**  
**Cause:** You specified the **-P** option without providing a prompt string as an argument.  
**Action:** Provide the missing prompt.
- Message:** *more: system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** No match found for regular expression.  
**Cause:** **more** did not find a match for the specified regular expression.  
**Action:** Check that the regular expression was entered correctly.

- 
- Message:** No remembered regular expression.  
**Cause:** You tried to use a remembered regular expression; however, there was no remembered regular expression.  
**Action:** Specify the regular expression explicitly.
- Message:** No such mark  
**Cause:** You attempted to move to a mark using the *'letter* command, but you never defined the mark *letter* with a **mletter** command.  
**Action:** Check the name of the mark to which you intended to move, and enter the correct name with the *'* command, or define a mark with the name specified.
- Message:** No tags file present.  
**Cause:** **more** was unable to open the tags file.  
**Action:** Check that the file tags exists, and that you have appropriate permissions. If tags does not exist, use the **ctags** command to create it.
- Message:** Number "*num*": *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).
- Message:** Syntax error in word expansion.  
**Cause:** You provided an invalid file name pattern.  
**Action:** Check the pattern and try again.
- Message:** Tag *tagname* not found.  
**Cause:** You tried to move to tag *tagname*, but **more** could not find it.  
**Action:** Check to see that you entered *tagname* correctly.
- Message:** Unknown option "*-option*"  
**Cause:** You specified an option that is not valid for **more**.  
**Action:** Check the *DESCRIPTION* section for a list of valid **more** options.
- Message:** window size too large  
**Cause:** You specified a window size (lines per page) with the *-* or **-n** option that was greater than the number of lines on the screen (as given by the environment variable *LINE*).  
**Action:** Specify a smaller window size.
- Message:** window size too small  
**Cause:** You specified a window size (lines per page) with the *-* or **-n** option that was less than three.  
**Action:** Specify a window size of at least three lines.



**PORTABILITY**

POSIX.2. x/OPEN Portability Guide 4.0. All UNIX systems.

The **-A**, **-P**, and **-S** options and the **:w** and **!** commands are extensions to the POSIX standard. The **HOME**, **END**, **PgDn**, **PgUp**, **↓**, and **↑** commands are extensions to traditional implementations of **more**, available only on terminal types which support these keys.

**MPE/iX NOTES**

The current MPE/iX implementation of **more** converts non-byte stream files to byte stream files before displaying them. File characteristics like file code, record size, and so forth are not preserved by this conversion. The output of **more** is written as a byte stream file.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

**SEE ALSO**

**cat(1)**, **ctags(1)**, **vi(1)**

**NAME**

mv — rename and move files and directories

**SYNOPSIS**

```
mv [-fi] file1 file2
```

```
mv [-fi] file ... directory
```

```
mv -R|-r [-fi] directory1 directory2
```

**DESCRIPTION**

**mv** renames files or moves them to a different directory. If you specify multiple *files*, the target (that is, the last path name on the command line) must be a directory. **mv** moves the files into that directory and gives them names that match the final components of the source path names. When you specify a single source *file* and the target is not a directory, **mv** moves the source to the new name, by a simple rename if possible.

If a destination file exists for which you do not have write permission, **mv** prompts with the name of the existing file. If you answer *y* or *yes*, it deletes the destination and then moves the source.

**Options**

**mv** accepts the following options:

- f** does not ask if you want to overwrite an existing destination without write permission; it automatically behaves as if you answered *yes*. If you specify both **-f** and **-i**, **mv** uses the option which appears last on the command line.
- i** always prompts before overwriting an existing file, whether or not the file is read-only. If you specify both **-f** and **-i**, **mv** uses the option which appears last on the command line.
- R** moves a directory and all its contents (files, subdirectories, files in subdirectories, and so on). For example,

```
mv -R dir1 dir2
```

moves the entire contents of *dir1* to *dir2/dir1*. **mv** creates any directories that it needs.
- r** is equivalent to **-R**.

---

**DIAGNOSTICS**

Possible exit status values are:

- 0 Successful completion.
- 1 Failure due to any of the following:
  - argument had trailing / but was not a directory
  - file could not be found
  - input file could not be opened for reading
  - output file could not be created or opened for output
  - read error occurred on an input file
  - write error occurred on an output file
  - input and output files were the same file
  - input file could not be unlinked
  - input file could not be renamed
  - fatal error was encountered when using the **-r** option

Possible fatal **-r** errors include the following:

  - inability to access a file
  - inability to read a directory
  - inability to remove a directory
  - inability to create a directory
  - a target which is not a directory
  - the source and destination directories are the same
- 2 Failure due to any of the following:
  - invalid command line option
  - too few arguments on the command line
  - a target that should be a directory but isn't
  - no space left on target device
  - out of memory to hold the data to be copied
  - the inability to create a directory to hold a target file

**Messages**

- Message:** cannot allocate target string  
**Cause:** **mv** has no space to hold the name of the target file.  
**Action:** Free up more system resources.
- Message:** cannot allocate I/O buffer: *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

---

**Message:** cannot create parent directory for target "*name*"  
**Cause:** An error occurred while trying to create the parent directory of the specified target file.  
**Action:** Make sure you have permissions to create the directory.

**Message:** cannot find file "*filename*"  
**Cause:** You specified a *filename* that does not exist.  
**Action:** Check the path and spelling of *filename*.

**Message:** cannot mkdir "*pathname*": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

**Message:** cannot open file "*filename*": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

**Message:** cannot rename "*file1*" to "*file2*": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

**Message:** Cannot reset permissions on file "*filename*": *system error*"  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

**Message:** Cannot reset times on file "*filename*": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

**Message:** Cannot reset uid or gid on file "*filename*": *system error*"  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

**Message:** cannot rmdir "*pathname*": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

**Message:** cannot unlink source file "*filename*": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

**Message:** fifo "*filename*": *system error*  
**Cause:** See **syserror(3)**.  
**Action:** See **syserror(3)**.

---

**Message:** no space on device for file "*filename*"  
**Cause:** You attempted to move a file to *filename* on a device that has no space for it.  
**Action:** Free up space on the target device or move the file to another device.

**Message:** read error on file "*filename*": *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).

**Message:** recursive copy to directory "*pathname*"  
**Cause:** You tried to recursively copy a directory to itself.  
**Action:** Choose a different *pathname*.

**Message:** source "*name*" and target "*name*" are identical  
**Cause:** You specified source and target files that are actually the same file (for example, because of links).  
**Action:** No further action is required.

**Message:** special file "*filename*" *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).

**Message:** stat error for "*filename*": *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).

**Message:** target file "*filename*": *system error*  
**Cause:** See **syserror**(3).  
**Action:** See **syserror**(3).

**Message:** target "*name*" must be a directory  
**Cause:** You attempted to move two or more files but the target indicated by *name* did not exist or was not a directory.  
**Action:** When moving two or more files, ensure that the final *name* on the command line is a directory.

**Message:** target "*pathname*" is not a directory  
**Cause:** When recursively moving directories with the **-r** or **-R** option, you specified a target which already existed, but was not a directory.  
**Action:** Check the spelling of the target *pathname*.

**Message:** Unknown option "*-option*"  
**Cause:** You specified an option that is not valid for **mv**.  
**Action:** Check the *DESCRIPTION* section of this man page for a list of valid **mv** options.

---

**Message:** unreadable directory "*pathname*": *system error*

**Cause:** See **syserror(3)**.

**Action:** See **syserror(3)**.

**Message:** write error on file "*filename*": *system error*

**Cause:** See **syserror(3)**.

**Action:** See **syserror(3)**.

### PORTABILITY

POSIX.2. *x*/OPEN Portability Guide 4.0. All UNIX systems.

The **-R** and **-r** options are extensions to the POSIX standard.

### MPE/iX NOTES

On MPE/iX, **mv** is available as both a built-in shell utility and an external utility.

For more information on how the current MPE/iX implementation may affect the operation of this utility, see Appendix A, *MPE/iX Implementation Considerations*.

### SEE ALSO

**cp(1)**, **cpio(1)**, **rm(1)**

# Permuted Index

---

|                               |          |                                           |               |
|-------------------------------|----------|-------------------------------------------|---------------|
| successfully/                 | :        | (colon) — do nothing, .....               | colon(1)      |
| in current environment/       | .        | (dot) — execute shell file .....          | dot(1)        |
| return from shell function or | .        | (dot) script .....                        | return(1)     |
| intercept                     |          | abnormal conditions and interrupts .....  | trap(1)       |
| display information           |          | about current users .....                 | who(1)        |
| display info                  |          | about RCS files .....                     | rlog(1)       |
| change                        |          | access permissions of a file .....        | chmod(1)      |
| split a text file,            |          | according to criteria .....               | csplit(1)     |
| command aliases/              | alias    | — display or create .....                 | alias(1)      |
| create a tracked              | alias    | .....                                     | hash(1)       |
| remove                        |          | alias definitions .....                   | unalias(1)    |
| display or create command     | aliases  | .....                                     | alias(1)      |
| write to                      |          | all logged in users .....                 | wall(1)       |
|                               |          | allow or refuse messages .....            | mesg(1)       |
| lexical                       |          | analyzer generator .....                  | lex(1)        |
| change the ownership of files |          | and/or directories .....                  | chown(1)      |
| the group ownership of files  |          | and/or directories/change .....           | chgrp(1)      |
| write to                      |          | another user .....                        | write(1)      |
|                               |          | appointment reminder system .....         | calendar(1)   |
| archives/                     | ar       | — create and maintain library .....       | ar(1)         |
|                               |          | arbitrary precision desk calculator ..... | dc(1)         |
| calculation language/         |          | arbitrary-precision arithmetic .....      | bc(1)         |
| file backup/                  |          | archiver for data interchange and .....   | pax(1)        |
|                               |          | archiver to copy and back up files .....  | cpio(1)       |
| USTAR-compatible tape         |          | archiver to copy and back up files .....  | tar(1)        |
| create and maintain library   | archives | .....                                     | ar(1)         |
| format of cpio                | archives | .....                                     | cpio(2)       |
| format of pax                 | archives | .....                                     | pax(2)        |
| format of tar                 | archives | .....                                     | tar(2)        |
| display                       |          | arguments .....                           | echo(1)       |
| display                       |          | arguments from the shell .....            | print(1)      |
| evaluate                      |          | arguments in shell .....                  | eval(1)       |
| arbitrary-precision           |          | arithmetic calculation language .....     | bc(1)         |
| evaluate                      |          | arithmetic expressions .....              | let(1)        |
| related books and             | articles | .....                                     | references(3) |
| table of                      |          | ASCII collating sequence .....            | ascii(3)      |
| carriage control/             | asa      | — interpret ASA/FORTRAN .....             | asa(1)        |
| interpret                     |          | ASA/FORTRAN carriage control .....        | asa(1)        |

|                                  |                                           |               |
|----------------------------------|-------------------------------------------|---------------|
| sequence/                        | ascii — table of ASCII collating .....    | ascii(3)      |
| variables/                       | assign attributes and values to .....     | typeset(1)    |
| run a command                    | at a different priority .....             | nice(1)       |
| assign                           | attributes and values to variables .....  | typeset(1)    |
| change RCS file                  | attributes .....                          | rcs(1)        |
| file and directory names and     | attributes/list .....                     | ls(1)         |
|                                  | Available code sets for conversion .....  | iconv(3)      |
| report generation language/      | awk — data transformation, .....          | awk(1)        |
| archiver to copy and             | back up files .....                       | cpio(1)       |
| tape archiver to copy and        | back up files/USTAR-compatible .....      | tar(1)        |
| for data interchange and file    | backup/archiver .....                     | pax(1)        |
| font/                            | banner — display text in large .....      | banner(1)     |
| component of path name/          | basename — display file name .....        | basename(1)   |
| files on a page-by-page          | basis/display .....                       | more(1)       |
| arithmetic calculation language/ | bc — arbitrary-precision .....            | bc(1)         |
| and show differences/            | bdiff — compare two text files .....      | diff(1)       |
| decode transmitted               | binary file .....                         | uudecode(1)   |
| compare                          | binary files and show differences .....   | diffb(1)      |
| display printable strings in     | binary files .....                        | strings(1)    |
| compute checksum and             | block count for file .....                | sum(1)        |
| copy and convert input           | blocks .....                              | dd(1)         |
| related                          | books and articles .....                  | references(3) |
| script/                          | break — exit from loop in shell .....     | break(1)      |
|                                  | break lines into shorter lines .....      | fold(1)       |
| display                          | brief command explanations .....          | help(1)       |
|                                  | build a special file .....                | mknod(1)      |
| compute checksum and             | byte count for file .....                 | cksum(1)      |
| convert MPE record files to      | byte stream files .....                   | tobyte(1)     |
| files/convert a                  | byte stream files to MPE record .....     | frombyte(1)   |
| count of newlines, words,        | bytes, and characters .....               | wc(1)         |
|                                  | c — produce multi-column output .....     | c(1)          |
| generic                          | C compiler interface .....                | c89(1)        |
| interface/                       | c89 — generic C compiler .....            | c89(1)        |
| arbitrary-precision arithmetic   | calculation language .....                | bc(1)         |
| arbitrary precision desk         | calculator .....                          | dc(1)         |
| system/                          | calendar — appointment reminder .....     | calendar(1)   |
| from the MPE/iX Shell/           | callci — run a MPE/iX CI command .....    | callci(1)     |
| interpret ASA/FORTRAN            | carriage control .....                    | asa(1)        |
| text files/                      | cat — concatenate and display .....       | cat(1)        |
| list file system elements in     | categories .....                          | lc(1)         |
|                                  | cd — change working directory .....       | cd(1)         |
|                                  | change access permissions of a file ..... | chmod(1)      |
|                                  | change file modification date .....       | touch(1)      |
|                                  | change file using diff output .....       | patch(1)      |
|                                  | change RCS file attributes .....          | rcs(1)        |



|                                        |                                                 |             |
|----------------------------------------|-------------------------------------------------|-------------|
|                                        | change terminal characteristics .....           | tput(1)     |
| files and/or directories/              | change the group ownership of.....              | chgrp(1)    |
| and/or directories/                    | change the ownership of files .....             | chown(1)    |
|                                        | change working directory .....                  | cd(1)       |
| reverse                                | character order of input lines.....             | rev(1)      |
| change terminal                        | characteristics .....                           | tput(1)     |
| selectively display fields or          | characters from input lines.....                | cut(1)      |
| of newlines, words, bytes, and         | characters/count .....                          | wc(1)       |
|                                        | check in a file under RCS.....                  | ci(1)       |
|                                        | check out a file under RCS .....                | co(1)       |
|                                        | check path names.....                           | pathchk(1)  |
| compute                                | checksum and block count for file.....          | sum(1)      |
| compute                                | checksum and byte count for file.....           | cksum(1)    |
| ownership of files and/or directories/ | chgrp — change the group.....                   | chgrp(1)    |
| permissions of a file/                 | chmod — change access .....                     | chmod(1)    |
| files and/or directories/              | chown — change the ownership of.....            | chown(1)    |
|                                        | ci — check in a file under RCS .....            | ci(1)       |
| run a MPE/iX                           | CI command from the MPE/iX Shell.....           | calci(1)    |
| byte count for file/                   | cksum — compute checksum and.....               | cksum(1)    |
|                                        | clean up working files .....                    | rcsclean(1) |
|                                        | clone output stream .....                       | tee(1)      |
|                                        | cmp — compare two files .....                   | cmp(1)      |
|                                        | co — check out a file under RCS .....           | co(1)       |
| Available                              | code sets for conversion.....                   | iconv(3)    |
| table of ASCII                         | collating sequence .....                        | ascii(3)    |
| show differences/                      | comm — compare sorted files and .....           | comm(1)     |
| command/                               | command — execute a simple.....                 | command(1)  |
| display or create                      | command aliases.....                            | alias(1)    |
| shell/interactive                      | command and history editing in the .....        | shedit(3)   |
| run a                                  | command at a different priority .....           | nice(1)     |
| edit and re-execute previous           | command .....                                   | r(1)        |
| execute a simple                       | command .....                                   | command(1)  |
| display brief                          | command explanations.....                       | help(1)     |
| run a MPE/iX CI                        | command from the MPE/iX Shell .....             | calci(1)    |
| display                                | command history .....                           | history(1)  |
| shell/execute a                        | command in place of the current .....           | exec(1)     |
| (Korn) shell and                       | command interpreter/POSIX-compliant .....       | sh(1)       |
| (Korn) shell and                       | command interpreter/rsh — POSIX-compliant ..... | sh(1)       |
| options from shell script              | command line/parse.....                         | getopts(1)  |
| construct and execute                  | command lines .....                             | xargs(1)    |
| tell how shell interprets              | command name .....                              | whence(1)   |
| external                               | command to parse shell file options .....       | getopt(1)   |
| path name for executable               | command/display .....                           | which(1)    |
| edit and re-enter previous             | commands/display, fix,.....                     | fc(1)       |
| differences/                           | compare binary files and show .....             | diffb(1)    |

|                                    |                                               |               |
|------------------------------------|-----------------------------------------------|---------------|
|                                    | compare RCS revisions .....                   | rcsdiff(1)    |
| differences/                       | compare sorted files and show .....           | comm(1)       |
|                                    | compare three text files .....                | diff3(1)      |
|                                    | compare two files .....                       | cmp(1)        |
| differences/                       | compare two text files and show .....         | diff(1)       |
| generic C                          | compiler interface .....                      | c89(1)        |
| wait for process to                | complete .....                                | wait(1)       |
| display file name                  | component of path name .....                  | basename(1)   |
| display directory                  | components of path name .....                 | dirname(1)    |
| compression of a file/             | compress — Lempel-Ziv .....                   | compress(1)   |
|                                    | compress files by Huffman encoding .....      | pack(1)       |
|                                    | compress spaces into tabs .....               | unexpand(1)   |
| Lempel-Ziv                         | compression of a file .....                   | compress(1)   |
| Undo Lempel-Ziv                    | compression of a file .....                   | uncompress(1) |
| for file/                          | compute checksum and block count .....        | sum(1)        |
| for file/                          | compute checksum and byte count .....         | cksum(1)      |
|                                    | concatenate and display text files .....      | cat(1)        |
| horizontally                       | concatenate lines .....                       | paste(1)      |
| test for                           | condition .....                               | test(1)       |
| intercept abnormal                 | conditions and interrupts .....               | trap(1)       |
| display POSIX                      | configuration information .....               | getconf(1)    |
|                                    | construct and execute command lines .....     | xargs(1)      |
| iteration of loop in shell script/ | continue — skip to next .....                 | continue(1)   |
| interpret ASA/FORTRAN carriage     | control .....                                 | asa(1)        |
| Available code sets for            | conversion .....                              | iconv(3)      |
| SCCS to RCS                        | conversion utility .....                      | sccs2rcs(1)   |
| record files/                      | convert a byte stream files to MPE .....      | frombyte(1)   |
| copy and                           | convert input blocks .....                    | dd(1)         |
| stream files/                      | convert MPE record files to byte .....        | tobyte(1)     |
| archiver to                        | copy and back up files .....                  | cpio(1)       |
| tape archiver to                   | copy and back up files/USTAR-compatible ..... | tar(1)        |
|                                    | copy and convert input blocks .....           | dd(1)         |
|                                    | copy files .....                              | cp(1)         |
|                                    | copy one line of standard input .....         | line(1)       |
| compute checksum and block         | count for file .....                          | sum(1)        |
| compute checksum and byte          | count for file .....                          | cksum(1)      |
| and characters/                    | count of newlines, words, bytes, .....        | wc(1)         |
|                                    | cp — copy files .....                         | cp(1)         |
| up files/                          | cpio — archiver to copy and back .....        | cpio(1)       |
|                                    | cpio — format of cpio archives .....          | cpio(2)       |
| format of                          | cpio archives .....                           | cpio(2)       |
|                                    | create a FIFO special file .....              | mkfifo(1)     |
|                                    | create a link to an existing file .....       | ln(1)         |
|                                    | create a new directory .....                  | mkdir(1)      |
|                                    | create a tracked alias .....                  | hash(1)       |

|                                 |                                         |             |
|---------------------------------|-----------------------------------------|-------------|
| archives/                       | create and maintain library .....       | ar(1)       |
| display or                      | create command aliases.....             | alias(1)    |
| get or set the file mode        | creation mask .....                     | umask(1)    |
| a text file, according to       | criteria/split .....                    | csplit(1)   |
| according to criteria/          | csplit — split a text file, .....       | csplit(1)   |
| ex, more, and vi/               | ctags — produce tags file for .....     | ctags(1)    |
| — execute shell file in         | current environment/. (dot).....        | dot(1)      |
| display status of jobs in       | current session .....                   | jobs(1)     |
| a command in place of the       | current shell/execute .....             | exec(1)     |
| display information about       | current users .....                     | who(1)      |
| or characters from input lines/ | cut — selectively display fields.....   | cut(1)      |
| archiver for                    | data interchange and file backup .....  | pax(1)      |
| generation language/            | data transformation, report.....        | awk(1)      |
| uncompress and display          | data .....                              | zcat(1)     |
| two sorted, textual relational  | databases/join.....                     | join(1)     |
| time/                           | date — set and display date and .....   | date(1)     |
| set and display                 | date and time .....                     | date(1)     |
| change file modification        | date .....                              | touch(1)    |
| calculator /                    | dc — arbitrary precision desk.....      | dc(1)       |
| blocks/                         | dd — copy and convert input .....       | dd(1)       |
| files/remove                    | debug information from executable ..... | strip(1)    |
|                                 | declare an integer variable .....       | integer(1)  |
|                                 | decode Huffman packed files .....       | unpack(1)   |
|                                 | decode transmitted binary file .....    | uudecode(1) |
| remove alias                    | definitions .....                       | unalias(1)  |
| mail                            | delivery program .....                  | tsmail(3)   |
| RCS editing facilities for      | descriptions and log messages.....      | rcsedit(3)  |
| arbitrary precision             | desk calculator .....                   | dc(1)       |
|                                 | determine file type .....               | file(1)     |
| and show differences/           | diff — compare two text files .....     | diff(1)     |
| change file using               | diff output.....                        | patch(1)    |
|                                 | diff3 — compare three text files.....   | diff3(1)    |
| show differences/               | diffb — compare binary files and.....   | diffb(1)    |
| compare binary files and show   | differences .....                       | diffb(1)    |
| compare sorted files and show   | differences .....                       | comm(1)     |
| two text files and show         | differences/bdiff — compare.....        | diff(1)     |
| two text files and show         | differences/compare .....               | diff(1)     |
| two text files and show         | differences/diffh — compare.....        | diff(1)     |
| run a command at a              | different priority .....                | nice(1)     |
| and show differences/           | diffh — compare two text files .....    | diff(1)     |
| rename and move files and       | directories .....                       | mv(1)       |
| ownership of files and/or       | directories/change the group .....      | chgrp(1)    |
| the ownership of files and/or   | directories/change .....                | chown(1)    |
| change working                  | directory .....                         | cd(1)       |
| display                         | directory components of path name ..... | dirname(1)  |

|                          |                                           |              |
|--------------------------|-------------------------------------------|--------------|
| create a new             | directory .....                           | mkdir(1)     |
| display working          | directory .....                           | pwd(1)       |
| list file and            | directory names and attributes .....      | ls(1)        |
| remove                   | directory .....                           | rmdir(1)     |
| components of path name/ | dirname — display directory .....         | dirname(1)   |
|                          | display a formatted string .....          | printf(1)    |
|                          | display and format files .....            | pr(1)        |
|                          | display arguments .....                   | echo(1)      |
|                          | display arguments from the shell .....    | print(1)     |
|                          | display brief command explanations .....  | help(1)      |
|                          | display command history .....             | history(1)   |
| uncompress and           | display data .....                        | zcat(1)      |
| set and                  | display date and time .....               | date(1)      |
| path name/               | display directory components of .....     | dirname(1)   |
| environment for process/ | display environment, set .....            | env(1)       |
| input lines/selectively  | display fields or characters from .....   | cut(1)       |
| path name/               | display file name component of .....      | basename(1)  |
| basis/                   | display files on a page-by-page .....     | more(1)      |
|                          | display first part of file .....          | head(1)      |
| previous commands/       | display, fix, edit and re-enter .....     | fc(1)        |
| standard output/         | display Huffman packed files on .....     | pcat(1)      |
|                          | display info about RCS files .....        | rlog(1)      |
| users/                   | display information about current .....   | who(1)       |
|                          | display last lines of file .....          | tail(1)      |
|                          | display or create command aliases .....   | alias(1)     |
|                          | display or modify shell functions .....   | functions(1) |
| command/                 | display path name for executable .....    | which(1)     |
| binary files/            | display printable strings in .....        | strings(1)   |
|                          | display process status .....              | ps(1)        |
| information/             | display POSIX configuration .....         | getconf(1)   |
| session/                 | display status of jobs in current .....   | jobs(1)      |
|                          | display system name information .....     | uname(1)     |
|                          | display terminal name .....               | tty(1)       |
| set or                   | display terminal options .....            | stty(1)      |
| concatenate and          | display text files .....                  | cat(1)       |
|                          | display text in large font .....          | banner(1)    |
|                          | display unique lines of sorted file ..... | uniq(1)      |
|                          | display user and group names .....        | id(1)        |
|                          | display user name .....                   | logname(1)   |
|                          | display working directory .....           | pwd(1)       |
| editor/                  | display-oriented interactive text .....   | vi(1)        |
|                          | do nothing, successfully .....            | true(1)      |
| from shell function or . | (dot) script/return .....                 | return(1)    |
|                          | du — summarize file space usage .....     | du(1)        |
| formatted file           | dump .....                                | od(1)        |

|                                   |                                           |             |
|-----------------------------------|-------------------------------------------|-------------|
|                                   | echo — display arguments.....             | echo(1)     |
|                                   | ed — line-oriented text editor .....      | ed(1)       |
| display, fix,                     | edit and re-enter previous commands ..... | fc(1)       |
| command/                          | edit and re-execute previous .....        | r(1)        |
| descriptions and log messages/RCS | editing facilities for .....              | rcsed(3)    |
| command and history               | editing in the shell/interactive .....    | shedit(3)   |
| line-oriented text                | editor.....                               | ed(1)       |
| stream                            | editor (non-interactive) .....            | sed(1)      |
| red — line-oriented text          | editor.....                               | ed(1)       |
| text                              | editor.....                               | ex(1)       |
| interactive text                  | editor/display-oriented .....             | vi(1)       |
|                                   | egrep — match patterns in a file.....     | grep(1)     |
| read                              | electronic mail .....                     | mailx(1)    |
| list file system                  | elements in categories .....              | lc(1)       |
|                                   | encode a file for safe transmission ..... | uuencode(1) |
| compress files by Huffman         | encoding .....                            | pack(1)     |
| environment for process/          | env — display environment, set .....      | env(1)      |
| variables/                        | environ — standard environment.....       | environ(3)  |
| execute shell file in current     | environment/. (dot) —.....                | dot(1)      |
| display environment, set          | environment for process.....              | env(1)      |
| process/display                   | environment, set environment for .....    | env(1)      |
| standard                          | environment variables .....               | environ(3)  |
| RCS                               | error messages.....                       | rcserror(3) |
| regular expression                | error messages.....                       | regerror(3) |
| system                            | error messages.....                       | syserror(3) |
| format of the                     | /etc/magic file .....                     | magic(2)    |
| shell/                            | eval — evaluate arguments in .....        | eval(1)     |
|                                   | evaluate arguments in shell.....          | eval(1)     |
|                                   | evaluate arithmetic expressions.....      | let(1)      |
|                                   | evaluate expression .....                 | expr(1)     |
|                                   | ex — text editor .....                    | ex(1)       |
| produce tags file for             | ex, more, and vi.....                     | ctags(1)    |
| place of the current shell/       | exec — execute a command in.....          | exec(1)     |
| display path name for             | executable command.....                   | which(1)    |
| remove debug information from     | executable files .....                    | strip(1)    |
| current shell/                    | execute a command in place of the .....   | exec(1)     |
|                                   | execute a simple command.....             | command(1)  |
| construct and                     | execute command lines .....               | xargs(1)    |
| suspend                           | execution for a specified time.....       | sleep(1)    |
| create a link to an               | existing file.....                        | ln(1)       |
|                                   | exit — exit from the shell .....          | exit(1)     |
|                                   | exit from loop in shell script .....      | break(1)    |
|                                   | exit from the shell.....                  | exit(1)     |
|                                   | expand — expand tabs to spaces .....      | expand(1)   |
|                                   | expand tabs to spaces .....               | expand(1)   |

|                                                          |                                        |             |
|----------------------------------------------------------|----------------------------------------|-------------|
| display brief command                                    | explanations .....                     | help(1)     |
|                                                          | export — mark names for export.....    | export(1)   |
| mark names for                                           | export.....                            | export(1)   |
|                                                          | expr — evaluate expression.....        | expr(1)     |
| regular                                                  | expression error messages.....         | regerror(3) |
| evaluate                                                 | expression.....                        | expr(1)     |
| syntax of regular                                        | expression patterns .....              | regex(3)    |
| evaluate arithmetic                                      | expressions .....                      | let(1)      |
| file options/<br>log messages/RCS editing                | external command to parse shell .....  | getopt(1)   |
|                                                          | facilities for descriptions and .....  | rcsedit(3)  |
|                                                          | fail, quietly .....                    | false(1)    |
|                                                          | false — fail, quietly .....            | false(1)    |
| re-enter previous commands/<br>lines/selectively display | fc — display, fix, edit and .....      | fc(1)       |
|                                                          | fgrep — match patterns in a file ..... | grep(1)     |
| create a                                                 | fields or characters from input .....  | cut(1)      |
|                                                          | FIFO special file .....                | mkfifo(1)   |
|                                                          | file — determine file type .....       | file(1)     |
| split a text                                             | file, according to criteria .....      | csplit(1)   |
| attributes/list                                          | file and directory names and .....     | ls(1)       |
| change RCS                                               | file attributes .....                  | rcs(1)      |
| for data interchange and                                 | file backup/archiver .....             | pax(1)      |
| build a special                                          | file .....                             | mknod(1)    |
| change access permissions of a                           | file .....                             | chmod(1)    |
| create a FIFO special                                    | file .....                             | mkfifo(1)   |
| create a link to an existing                             | file .....                             | ln(1)       |
| decode transmitted binary                                | file .....                             | uudecode(1) |
| display first part of                                    | file .....                             | head(1)     |
| display last lines of                                    | file .....                             | tail(1)     |
| display unique lines of sorted                           | file .....                             | uniq(1)     |
| formatted                                                | file dump.....                         | od(1)       |
| egrep — match patterns in a                              | file .....                             | grep(1)     |
| fgrep — match patterns in a                              | file .....                             | grep(1)     |
| produce tags                                             | file for ex, more, and vi .....        | ctags(1)    |
| encode a                                                 | file for safe transmission .....       | uencode(1)  |
| format of tags                                           | file .....                             | tags(2)     |
| format of RCS                                            | file .....                             | rcsfile(3)  |
| format of the /etc/magic                                 | file .....                             | magic(2)    |
| . (dot) — execute shell                                  | file in current environment.....       | dot(1)      |
| split a                                                  | file into manageable pieces.....       | split(1)    |
| Lempel-Ziv compression of a                              | file .....                             | compress(1) |
| look for keywords in a                                   | file .....                             | ident(1)    |
| match patterns in a                                      | file .....                             | grep(1)     |
| three-way                                                | file merge.....                        | merge(1)    |
| get or set the                                           | file mode creation mask .....          | umask(1)    |
| change                                                   | file modification date .....           | touch(1)    |

|                               |                                          |               |
|-------------------------------|------------------------------------------|---------------|
| display                       | file name component of path name .....   | basename(1)   |
| command to parse shell        | file options/external .....              | getopt(1)     |
| summarize                     | file space usage .....                   | du(1)         |
| list                          | file system elements in categories ..... | lc(1)         |
| find files within             | file tree .....                          | find(1)       |
| determine                     | file type .....                          | file(1)       |
| check in a                    | file under RCS .....                     | ci(1)         |
| check out a                   | file under RCS .....                     | co(1)         |
| change                        | file using diff output .....             | patch(1)      |
| checksum and block count for  | file/compute .....                       | sum(1)        |
| checksum and byte count for   | file/compute .....                       | cksum(1)      |
| rename and move               | files and directories .....              | mv(1)         |
| bdiff — compare two text      | files and show differences .....         | diff(1)       |
| compare binary                | files and show differences .....         | diffb(1)      |
| compare sorted                | files and show differences .....         | comm(1)       |
| compare two text              | files and show differences .....         | diff(1)       |
| diffh — compare two text      | files and show differences .....         | diff(1)       |
| change the group ownership of | files and/or directories .....           | chgrp(1)      |
| change the ownership of       | files and/or directories .....           | chown(1)      |
| archiver to copy and back up  | files .....                              | cpio(1)       |
| compress                      | files by Huffman encoding .....          | pack(1)       |
| clean up working              | files .....                              | rcsclean(1)   |
| compare three text            | files .....                              | diff3(1)      |
| compare two                   | files .....                              | cmp(1)        |
| concatenate and display text  | files .....                              | cat(1)        |
| copy                          | files .....                              | cp(1)         |
| decode Huffman packed         | files .....                              | unpack(1)     |
| display and format            | files .....                              | pr(1)         |
| display info about RCS        | files .....                              | rlog(1)       |
| display                       | files on a page-by-page basis .....      | more(1)       |
| display Huffman packed        | files on standard output .....           | pcat(1)       |
| remove                        | files .....                              | rm(1)         |
| convert MPE record            | files to byte stream files .....         | tobyte(1)     |
| convert a byte stream         | files to MPE record files .....          | frombyte(1)   |
| find                          | files within file tree .....             | find(1)       |
| stream files to MPE record    | files/convert a byte .....               | frombyte(1)   |
| record files to byte stream   | files/convert MPE .....                  | tobyte(1)     |
| printable strings in binary   | files/display .....                      | strings(1)    |
| and interdependent            | files/maintain program-generated .....   | make(1)       |
| and interdependent            | files/maintain program-generated .....   | make(1)       |
| information from executable   | files/remove debug .....                 | strip(1)      |
| archiver to copy and back up  | files/USTAR-compatible tape .....        | tar(1)        |
| Lempel-Ziv compression of a   | file/Undo .....                          | uncompress(1) |
| translation                   | filter .....                             | tr(1)         |
| tree/                         | find — find files within file .....      | find(1)       |

|                             |                                         |              |
|-----------------------------|-----------------------------------------|--------------|
|                             | find files within file tree .....       | find(1)      |
| display                     | first part of file.....                 | head(1)      |
| commands/display,           | fix, edit and re-enter previous .....   | fc(1)        |
| set shell                   | flags and positional parameters .....   | set(1)       |
|                             | fmt — simple text formatter .....       | fmt(1)       |
| lines/                      | fold — break lines into shorter.....    | fold(1)      |
| display text in large       | font.....                               | banner(1)    |
| display and                 | format files .....                      | pr(1)        |
|                             | format of cpio archives .....           | cpio(2)      |
|                             | format of tags file.....                | tags(2)      |
|                             | format of pax archives.....             | pax(2)       |
|                             | format of RCS file.....                 | rcsfile(3)   |
|                             | format of tar archives .....            | tar(2)       |
|                             | format of the /etc/magic file.....      | magic(2)     |
|                             | formatted file dump .....               | od(1)        |
| display a                   | formatted string.....                   | printf(1)    |
| simple text                 | formatter .....                         | fmt(1)       |
| files to MPE record files/  | frombyte — convert a byte stream.....   | frombyte(1)  |
| return from shell           | function or . (dot) script .....        | return(1)    |
| remove shell variable or    | function .....                          | unset(1)     |
| shell functions/            | functions — display or modify.....      | functions(1) |
| display or modify shell     | functions .....                         | functions(1) |
|                             | functions used with lex .....           | lex(3)       |
| data transformation, report | generation language .....               | awk(1)       |
| parser                      | generator language .....                | yacc(1)      |
| lexical analyzer            | generator.....                          | lex(1)       |
|                             | generic C compiler interface .....      | c89(1)       |
| mask/                       | get or set the file mode creation.....  | umask(1)     |
| configuration information/  | getconf — display POSIX.....            | getconf(1)   |
| parse shell file options/   | getopt — external command to .....      | getopt(1)    |
| shell script command line/  | getopts — parse options from.....       | getopts(1)   |
|                             | grep — match patterns in a file.....    | grep(1)      |
| display user and            | group names .....                       | id(1)        |
| directories/change the      | group ownership of files and/or .....   | chgrp(1)     |
|                             | hash — create a tracked alias .....     | hash(1)      |
|                             | head — display first part of file ..... | head(1)      |
| explanations/               | help — display brief command .....      | help(1)      |
|                             | history — display command history ..... | history(1)   |
| display command             | history .....                           | history(1)   |
| interactive command and     | history editing in the shell.....       | shedit(3)    |
|                             | horizontally concatenate lines .....    | paste(1)     |
| tell                        | how shell interprets command name.....  | whence(1)    |
| tell                        | how shell interprets name .....         | type(1)      |
| compress files by           | Huffman encoding .....                  | pack(1)      |
| decode                      | Huffman packed files .....              | unpack(1)    |



|                                   |                                         |             |
|-----------------------------------|-----------------------------------------|-------------|
| output/display                    | Huffman packed files on standard .....  | pcat(1)     |
| conversion/                       | iconv — Available code sets for .....   | iconv(3)    |
|                                   | id — display user and group names ..... | id(1)       |
| file/                             | ident — look for keywords in a .....    | ident(1)    |
| display                           | info about RCS files .....              | rlog(1)     |
| display                           | information about current users .....   | who(1)      |
| display POSIX configuration       | information .....                       | getconf(1)  |
| display system name               | information .....                       | uname(1)    |
| remove debug                      | information from executable files ..... | strip(1)    |
|                                   | input a line to the shell .....         | read(1)     |
| copy and convert                  | input blocks .....                      | dd(1)       |
| copy one line of standard         | input .....                             | line(1)     |
| reverse character order of        | input lines .....                       | rev(1)      |
| fields or characters from         | input lines/selectively display .....   | cut(1)      |
| variable/                         | integer — declare an integer .....      | integer(1)  |
| declare an                        | integer variable .....                  | integer(1)  |
| editing in the shell/             | interactive command and history .....   | shedit(3)   |
| display-oriented                  | interactive text editor .....           | vi(1)       |
| interrupts/                       | intercept abnormal conditions and ..... | trap(1)     |
| archiver for data                 | interchange and file backup .....       | pax(1)      |
| maintain program-generated and    | interdependent files .....              | make(1)     |
| generic C compiler                | interface .....                         | c89(1)      |
| control/                          | interpret ASA/FORTRAN carriage .....    | asa(1)      |
| (Korn) shell and command          | interpreter/POSIX-compliant .....       | sh(1)       |
| (Korn) shell and command          | interpreter/rsh — POSIX-compliant ..... | sh(1)       |
| tell how shell                    | interprets command name .....           | whence(1)   |
| tell how shell                    | interprets name .....                   | type(1)     |
| abnormal conditions and           | interrupts/intercept .....              | trap(1)     |
|                                   | intro — introduction to man pages ..... | intro(1)    |
|                                   | introduction to man pages .....         | intro(1)    |
| skip to next                      | iteration of loop in shell script ..... | continue(1) |
| current session/                  | jobs — display status of jobs in .....  | jobs(1)     |
| display status of                 | jobs in current session .....           | jobs(1)     |
| relational databases/             | join — join two sorted, textual .....   | join(1)     |
| relational databases/             | join two sorted, textual .....          | join(1)     |
| look for                          | keywords in a file .....                | ident(1)    |
|                                   | kill — terminate process .....          | kill(1)     |
| interpreter/POSIX-compliant       | (Korn) shell and command .....          | sh(1)       |
| interpreter/rsh — POSIX-compliant | (Korn) shell and command .....          | sh(1)       |
| parser generator                  | language .....                          | yacc(1)     |
| arithmetic calculation            | language/arbitrary-precision .....      | bc(1)       |
| report generation                 | language/data transformation, .....     | awk(1)      |
| display text in                   | large font .....                        | banner(1)   |
| display                           | last lines of file .....                | tail(1)     |
| in categories/                    | lc — list file system elements .....    | lc(1)       |

|                               |                                           |               |
|-------------------------------|-------------------------------------------|---------------|
|                               | Lempel-Ziv compression of a file.....     | compress(1)   |
| Undo                          | Lempel-Ziv compression of a file.....     | uncompress(1) |
| expressions/                  | let — evaluate arithmetic .....           | let(1)        |
|                               | lex — functions used with lex .....       | lex(3)        |
|                               | lex — lexical analyzer generator .....    | lex(1)        |
| functions used with           | lex .....                                 | lex(3)        |
|                               | lexical analyzer generator .....          | lex(1)        |
| create and maintain           | library archives .....                    | ar(1)         |
| input/                        | line — copy one line of standard .....    | line(1)       |
| copy one                      | line of standard input.....               | line(1)       |
| input a                       | line to the shell.....                    | read(1)       |
|                               | line-oriented text editor.....            | ed(1)         |
| from shell script command     | line/parse options.....                   | getopts(1)    |
| break lines into shorter      | lines.....                                | fold(1)       |
| construct and execute command | lines.....                                | xargs(1)      |
| horizontally concatenate      | lines.....                                | paste(1)      |
| break                         | lines into shorter lines .....            | fold(1)       |
| number                        | lines.....                                | nl(1)         |
| display last                  | lines of file.....                        | tail(1)       |
| display unique                | lines of sorted file .....                | uniq(1)       |
| character order of input      | lines/reverse .....                       | rev(1)        |
| or characters from input      | lines/selectively display fields.....     | cut(1)        |
| create a                      | link to an existing file.....             | ln(1)         |
| attributes/                   | list file and directory names and.....    | ls(1)         |
| categories/                   | list file system elements in .....        | lc(1)         |
| existing file/                | ln — create a link to an .....            | ln(1)         |
| setting                       | local time zone .....                     | timezone(3)   |
| for descriptions and          | log messages/RCS editing facilities ..... | rcsedit(3)    |
| write to all                  | logged in users .....                     | wall(1)       |
|                               | logname — display user name.....          | logname(1)    |
|                               | look for keywords in a file .....         | ident(1)      |
| exit from                     | loop in shell script .....                | break(1)      |
| skip to next iteration of     | loop in shell script .....                | continue(1)   |
| names and attributes/         | ls — list file and directory .....        | ls(1)         |
| file/                         | magic — format of the /etc/magic .....    | magic(2)      |
|                               | mail delivery program .....               | tmail(3)      |
| read electronic               | mail .....                                | mailx(1)      |
|                               | mailx — read electronic mail .....        | mailx(1)      |
| create and                    | maintain library archives .....           | ar(1)         |
| interdependent files/         | maintain program-generated and.....       | make(1)       |
| online reference manual/      | man — print sections of the.....          | man(1)        |
| introduction to               | man pages.....                            | intro(1)      |
| split a file into             | manageable pieces .....                   | split(1)      |
| of the online reference       | manual/print sections .....               | man(1)        |
|                               | mark names for export.....                | export(1)     |

|                                |                                         |              |
|--------------------------------|-----------------------------------------|--------------|
|                                | mark variable as readonly .....         | readonly(1)  |
| or set the file mode creation  | mask/get .....                          | umask(1)     |
|                                | match patterns in a file .....          | grep(1)      |
|                                | merge — three-way file merge .....      | merge(1)     |
|                                | merge RCS revisions .....               | rcsmerge(1)  |
| three-way file                 | merge .....                             | merge(1)     |
|                                | mesg — allow or refuse messages .....   | mesg(1)      |
| allow or refuse                | messages .....                          | mesg(1)      |
| RCS error                      | messages .....                          | rcserror(3)  |
| regular expression error       | messages .....                          | regerror(3)  |
| system error                   | messages .....                          | syserror(3)  |
| for descriptions and log       | messages/RCS editing facilities .....   | rcsedit(3)   |
|                                | mkdir — create a new directory .....    | mkdir(1)     |
| file/                          | mkfifo — create a FIFO special .....    | mkfifo(1)    |
|                                | mknod — build a special file .....      | mknod(1)     |
| get or set the file            | mode creation mask .....                | umask(1)     |
| change file                    | modification date .....                 | touch(1)     |
| display or                     | modify shell functions .....            | functions(1) |
| page-by-page basis /           | more — display files on a .....         | more(1)      |
| produce tags file for ex,      | more, and vi .....                      | ctags(1)     |
| rename and                     | move files and directories .....        | mv(1)        |
| convert a byte stream files to | MPE record files .....                  | frombyte(1)  |
| files/convert                  | MPE record files to byte stream .....   | tobyte(1)    |
| Shell/run a                    | MPE/iX CI command from the MPE/iX ..... | callci(1)    |
| a MPE/iX CI command from the   | MPE/iX Shell/run .....                  | callci(1)    |
| produce                        | multi-column output .....               | c(1)         |
| directories/                   | mv — rename and move files and .....    | mv(1)        |
| display file                   | name component of path name .....       | basename(1)  |
| display terminal               | name .....                              | tty(1)       |
| display user                   | name .....                              | logname(1)   |
| display path                   | name for executable command .....       | which(1)     |
| display system                 | name information .....                  | uname(1)     |
| tell how shell interprets      | name .....                              | type(1)      |
| directory components of path   | name/display .....                      | dirname(1)   |
| file name component of path    | name/display .....                      | basename(1)  |
| list file and directory        | names and attributes .....              | ls(1)        |
| check path                     | names .....                             | pathchk(1)   |
| display user and group         | names .....                             | id(1)        |
| mark                           | names for export .....                  | export(1)    |
| how shell interprets command   | name/tell .....                         | whence(1)    |
| create a                       | new directory .....                     | mkdir(1)     |
| characters/count of            | newlines, words, bytes, and .....       | wc(1)        |
| script/skip to                 | next iteration of loop in shell .....   | continue(1)  |
| different priority/            | nice — run a command at a .....         | nice(1)      |
|                                | nl — number lines .....                 | nl(1)        |

|                                                               |             |
|---------------------------------------------------------------|-------------|
| stream editor (non-interactive) .....                         | sed(1)      |
| : (colon) — do nothing, successfully .....                    | colon(1)    |
| do nothing, successfully .....                                | true(1)     |
| number lines.....                                             | nl(1)       |
| od — formatted file dump.....                                 | od(1)       |
| print sections of the online reference manual .....           | man(1)      |
| line/parse options from shell script command .....            | getopts(1)  |
| set or display terminal options .....                         | stty(1)     |
| command to parse shell file options/external .....            | getopt(1)   |
| reverse character order of input lines .....                  | rev(1)      |
| check out a file under RCS .....                              | co(1)       |
| change file using diff output.....                            | patch(1)    |
| produce multi-column output.....                              | c(1)        |
| clone output stream.....                                      | tee(1)      |
| packed files on standard output/display Huffman .....         | pcat(1)     |
| directories/change the ownership of files and/or.....         | chown(1)    |
| directories/change the group ownership of files and/or.....   | chgrp(1)    |
| encoding/ pack — compress files by Huffman .....              | pack(1)     |
| decode Huffman packed files.....                              | unpack(1)   |
| display Huffman packed files on standard output .....         | pcat(1)     |
| display files on a page-by-page basis .....                   | more(1)     |
| introduction to man pages .....                               | intro(1)    |
| set shell flags and positional parameters .....               | set(1)      |
| shift positional parameters .....                             | shift(1)    |
| command line/ parse options from shell script.....            | getopts(1)  |
| external command to parse shell file options .....            | getopt(1)   |
| parser generator language .....                               | yacc(1)     |
| display first part of file .....                              | head(1)     |
| lines/ paste — horizontally concatenate .....                 | paste(1)    |
| output/ patch — change file using diff.....                   | patch(1)    |
| display file name component of path name .....                | basename(1) |
| display path name for executable command.....                 | which(1)    |
| directory components of path name/display .....               | dirname(1)  |
| check path names.....                                         | pathchk(1)  |
| pathchk — check path names .....                              | pathchk(1)  |
| egrep — match patterns in a file.....                         | grep(1)     |
| fgrep — match patterns in a file.....                         | grep(1)     |
| match patterns in a file.....                                 | grep(1)     |
| syntax of regular expression patterns .....                   | regex(3)    |
| interchange and file backup/ pax — archiver for data .....    | pax(1)      |
| pax — format of pax archives .....                            | pax(2)      |
| format of pax archives .....                                  | pax(2)      |
| files on standard output/ pcat — display Huffman packed ..... | pcat(1)     |
| change access permissions of a file.....                      | chmod(1)    |
| split a file into manageable pieces .....                     | split(1)    |

|                                    |                                         |             |
|------------------------------------|-----------------------------------------|-------------|
| execute a command in               | place of the current shell .....        | exec(1)     |
| set shell flags and                | positional parameters .....             | set(1)      |
| shift                              | positional parameters .....             | shift(1)    |
| arbitrary                          | pr — display and format files .....     | pr(1)       |
| edit and re-execute                | precision desk calculator .....         | dc(1)       |
| fix, edit and re-enter             | previous command .....                  | r(1)        |
| the shell/                         | previous commands/display .....         | fc(1)       |
| reference manual/                  | print — display arguments from .....    | print(1)    |
| display                            | print sections of the online .....      | man(1)      |
| string/                            | printable strings in binary files ..... | strings(1)  |
| set                                | printf — display a formatted .....      | printf(1)   |
| run a command at a different       | priorities of running processes .....   | renice(1)   |
| display                            | priority .....                          | nice(1)     |
| terminate                          | process status .....                    | ps(1)       |
| wait for                           | process .....                           | kill(1)     |
| set environment for                | process to complete .....               | wait(1)     |
| set priorities of running          | process/display environment, .....      | env(1)      |
| and vi/                            | processes .....                         | renice(1)   |
| mail delivery                      | produce multi-column output .....       | c(1)        |
| interdependent files/maintain      | produce tags file for ex, more, .....   | ctags(1)    |
| display                            | program .....                           | tsmail(3)   |
| command interpreter/               | program-generated and .....             | make(1)     |
| fail,                              | ps — display process status .....       | ps(1)       |
| command/                           | pwd — display working directory .....   | pwd(1)      |
| check in a file under              | POSIX configuration information .....   | getconf(1)  |
| check out a file under             | POSIX-compliant (Korn) shell and .....  | sh(1)       |
| SCCS to                            | quietly .....                           | false(1)    |
| descriptions and log messages/     | r — edit and re-execute previous .....  | r(1)        |
| change                             | rcs — change RCS file attributes .....  | r(1)        |
| format of                          | RCS .....                               | ci(1)       |
| display info about                 | RCS .....                               | co(1)       |
| compare                            | RCS conversion utility .....            | scs2rcs(1)  |
| merge                              | RCS editing facilities for .....        | rcsedit(3)  |
| for descriptions and log messages/ | RCS error messages .....                | rcserror(3) |
| change                             | RCS file attributes .....               | rcs(1)      |
| format of                          | RCS file .....                          | rcsfile(3)  |
| display info about                 | RCS files .....                         | rlog(1)     |
| compare                            | RCS revisions .....                     | rcsdiff(1)  |
| merge                              | RCS revisions .....                     | rcsmerge(1) |
| for descriptions and log messages/ | rcsclean — clean up working files ..... | rcsclean(1) |
| change                             | rcsdiff — compare RCS revisions .....   | rcsdiff(1)  |
| format of                          | rcsedit — RCS editing facilities .....  | rcsedit(3)  |
| display info about                 | rcserror — RCS error messages .....     | rcserror(3) |
| compare                            | rcsfile — format of RCS file .....      | rcsfile(3)  |
| merge                              | rcsmerge — merge RCS revisions .....    | rcsmerge(1) |

|                                |                                         |               |
|--------------------------------|-----------------------------------------|---------------|
|                                | read — input a line to the shell.....   | read(1)       |
|                                | read electronic mail .....              | mailx(1)      |
| readonly/                      | readonly — mark variable as .....       | readonly(1)   |
| mark variable as               | readonly .....                          | readonly(1)   |
| convert MPE                    | record files to byte stream files ..... | tobyte(1)     |
| a byte stream files to MPE     | record files/convert .....              | frombyte(1)   |
|                                | red — line-oriented text editor .....   | ed(1)         |
| display, fix, edit and         | re-enter previous commands .....        | fc(1)         |
| edit and                       | re-execute previous command .....       | r(1)          |
| print sections of the online   | reference manual .....                  | man(1)        |
| articles/                      | references — related books and .....    | references(3) |
| allow or                       | refuse messages .....                   | msg(1)        |
| error messages/                | regerror — regular expression .....     | regerror(3)   |
| expression patterns/           | regexp — syntax of regular .....        | regexp(3)     |
|                                | regular expression error messages.....  | regerror(3)   |
| syntax of                      | regular expression patterns .....       | regexp(3)     |
|                                | related books and articles .....        | references(3) |
| join two sorted, textual       | relational databases .....              | join(1)       |
| appointment                    | reminder system.....                    | calendar(1)   |
|                                | remove alias definitions .....          | unalias(1)    |
| executable files/              | remove debug information from.....      | strip(1)      |
|                                | remove directory .....                  | rmdir(1)      |
|                                | remove files .....                      | rm(1)         |
|                                | remove shell variable or function ..... | unset(1)      |
| directories/                   | rename and move files and.....          | mv(1)         |
| running processes/             | renice — set priorities of .....        | renice(1)     |
| data transformation,           | report generation language .....        | awk(1)        |
| function or . (dot) script/    | return — return from shell .....        | return(1)     |
| (dot) script/                  | return from shell function or .....     | return(1)     |
| input lines/                   | rev — reverse character order of.....   | rev(1)        |
| lines/                         | reverse character order of input.....   | rev(1)        |
| compare RCS                    | revisions .....                         | rcsdiff(1)    |
| merge RCS                      | revisions .....                         | rcsmerge(1)   |
| files/                         | rlog — display info about RCS .....     | rlog(1)       |
|                                | rm — remove files .....                 | rm(1)         |
|                                | rmdir — remove directory .....          | rmdir(1)      |
| shell and command interpreter/ | rsh — POSIX-compliant (Korn).....       | sh(1)         |
| priority/                      | run a command at a different .....      | nice(1)       |
| MPE/iX Shell/                  | run a MPE/iX CI command from the .....  | callci(1)     |
| set priorities of              | running processes .....                 | renice(1)     |
| encode a file for              | safe transmission.....                  | uuencode(1)   |
|                                | SCCS to RCS conversion utility .....    | sccs2rcs(1)   |
| conversion utility/            | sccs2rcs — SCCS to RCS .....            | sccs2rcs(1)   |
| parse options from shell       | script command line .....               | getopts(1)    |
| exit from loop in shell        | script .....                            | break(1)      |

|                                |                                           |              |
|--------------------------------|-------------------------------------------|--------------|
| shell function or . (dot)      | script/return from.....                   | return(1)    |
| iteration of loop in shell     | script/skip to next .....                 | continue(1)  |
| manual/print                   | sections of the online reference .....    | man(1)       |
| (non-interactive)/             | sed — stream editor.....                  | sed(1)       |
| characters from input lines/   | selectively display fields or.....        | cut(1)       |
| table of ASCII collating       | sequence .....                            | ascii(3)     |
| status of jobs in current      | session/display .....                     | jobs(1)      |
| positional parameters/         | set — set shell flags and.....            | set(1)       |
|                                | set and display date and time .....       | date(1)      |
| display environment,           | set environment for process .....         | env(1)       |
|                                | set or display terminal options.....      | stty(1)      |
|                                | set priorities of running processes ..... | renice(1)    |
| parameters/                    | set shell flags and positional .....      | set(1)       |
|                                | set terminal tab stops.....               | tabs(1)      |
| get or                         | set the file mode creation mask .....     | umask(1)     |
| Available code                 | sets for conversion .....                 | iconv(3)     |
|                                | setting local time zone .....             | timezone(3)  |
| shell and command interpreter/ | sh — POSIX-compliant (Korn) .....         | sh(1)        |
| history editing in the shell/  | shedit — interactive command and.....     | shedit(3)    |
| POSIX-compliant (Korn)         | shell and command interpreter .....       | sh(1)        |
| rsh — POSIX-compliant (Korn)   | shell and command interpreter .....       | sh(1)        |
| display arguments from the     | shell.....                                | print(1)     |
| evaluate arguments in          | shell.....                                | eval(1)      |
| exit from the                  | shell.....                                | exit(1)      |
| . (dot) — execute              | shell file in current environment.....    | dot(1)       |
| external command to parse      | shell file options .....                  | getopt(1)    |
| parameters/set                 | shell flags and positional .....          | set(1)       |
| return from                    | shell function or . (dot) script .....    | return(1)    |
| display or modify              | shell functions .....                     | functions(1) |
| input a line to the            | shell.....                                | read(1)      |
| tell how                       | shell interprets command name .....       | whence(1)    |
| tell how                       | shell interprets name .....               | type(1)      |
| parse options from             | shell script command line .....           | getopts(1)   |
| exit from loop in              | shell script .....                        | break(1)     |
| to next iteration of loop in   | shell script/skip .....                   | continue(1)  |
| remove                         | shell variable or function .....          | unset(1)     |
| in place of the current        | shell/execute a command .....             | exec(1)      |
| and history editing in the     | shell/interactive command.....            | shedit(3)    |
| CI command from the MPE/iX     | Shell/run a MPE/iX .....                  | callci(1)    |
| parameters/                    | shift — shift positional.....             | shift(1)     |
|                                | shift positional parameters .....         | shift(1)     |
| break lines into               | shorter lines .....                       | fold(1)      |
| compare binary files and       | show differences .....                    | diffb(1)     |
| compare sorted files and       | show differences .....                    | comm(1)      |
| compare two text files and     | show differences .....                    | diff(1)      |

|                              |                                           |             |
|------------------------------|-------------------------------------------|-------------|
| — compare two text files and | show differences/bdiff .....              | diff(1)     |
| — compare two text files and | show differences/diffh .....              | diff(1)     |
| execute a                    | simple command .....                      | command(1)  |
|                              | simple text formatter .....               | fmt(1)      |
| shell script/                | skip to next iteration of loop in .....   | continue(1) |
| specified time/              | sleep — suspend execution for a .....     | sleep(1)    |
|                              | sort — sort/merge utility .....           | sort(1)     |
| display unique lines of      | sorted file .....                         | uniq(1)     |
| compare                      | sorted files and show differences .....   | comm(1)     |
| databases/join two           | sorted, textual relational .....          | join(1)     |
|                              | sort/merge utility .....                  | sort(1)     |
| summarize file               | space usage .....                         | du(1)       |
| expand tabs to               | spaces .....                              | expand(1)   |
| compress                     | spaces into tabs .....                    | unexpand(1) |
| build a                      | special file .....                        | mknod(1)    |
| create a FIFO                | special file .....                        | mkfifo(1)   |
| suspend execution for a      | specified time .....                      | sleep(1)    |
| manageable pieces/           | split — split a file into .....           | split(1)    |
|                              | split a file into manageable pieces ..... | split(1)    |
| criteria/                    | split a text file, according to .....     | csplit(1)   |
|                              | standard environment variables .....      | envron(3)   |
| copy one line of             | standard input .....                      | line(1)     |
| Huffman packed files on      | standard output/display .....             | pcat(1)     |
| display process              | status .....                              | ps(1)       |
| display                      | status of jobs in current session .....   | jobs(1)     |
| set terminal tab             | stops .....                               | tabs(1)     |
| clone output                 | stream .....                              | tee(1)      |
|                              | stream editor (non-interactive) .....     | sed(1)      |
| convert a byte               | stream files to MPE record files .....    | frombyte(1) |
| MPE record files to byte     | stream files/convert .....                | tobyte(1)   |
| display a formatted          | string .....                              | printf(1)   |
| strings in binary files/     | strings — display printable .....         | strings(1)  |
| display printable            | strings in binary files .....             | strings(1)  |
| from executable files/       | strip — remove debug information .....    | strip(1)    |
| options/                     | stty — set or display terminal .....      | stty(1)     |
| : (colon) — do nothing,      | successfully .....                        | colon(1)    |
| do nothing,                  | successfully .....                        | true(1)     |
| count for file/              | sum — compute checksum and block .....    | sum(1)      |
|                              | summarize file space usage .....          | du(1)       |
| time/                        | suspend execution for a specified .....   | sleep(1)    |
| patterns/                    | syntax of regular expression .....        | regex(3)    |
|                              | syserror — system error messages .....    | syserror(3) |
| appointment reminder         | system .....                              | calendar(1) |
| list file                    | system elements in categories .....       | lc(1)       |
|                              | system error messages .....               | syserror(3) |



|                                      |                        |                                         |             |
|--------------------------------------|------------------------|-----------------------------------------|-------------|
|                                      | display                | system name information .....           | uname(1)    |
|                                      | set terminal           | tab stops.....                          | tabs(1)     |
|                                      |                        | table of ASCII collating sequence.....  | ascii(3)    |
|                                      |                        | tabs — set terminal tab stops .....     | tabs(1)     |
|                                      | compress spaces into   | tabs.....                               | unexpand(1) |
|                                      | expand                 | tabs to spaces .....                    | expand(1)   |
|                                      |                        | tags — format of tags file.....         | tags(2)     |
|                                      | format of              | tags file .....                         | tags(2)     |
|                                      | produce                | tags file for ex, more, and vi .....    | ctags(1)    |
|                                      |                        | tail — display last lines of file.....  | tail(1)     |
|                                      | files/USTAR-compatible | tape archiver to copy and back up.....  | tar(1)      |
|                                      |                        | tar — format of tar archives.....       | tar(2)      |
| archiver to copy and back up files / |                        | tar — USTAR-compatible tape .....       | tar(1)      |
|                                      | format of              | tar archives .....                      | tar(2)      |
|                                      |                        | tee — clone output stream .....         | tee(1)      |
|                                      | name/                  | tell how shell interprets command ..... | whence(1)   |
|                                      |                        | tell how shell interprets name.....     | type(1)     |
|                                      | change                 | terminal characteristics .....          | tput(1)     |
|                                      | display                | terminal name .....                     | tty(1)      |
|                                      | set or display         | terminal options .....                  | stty(1)     |
|                                      | set                    | terminal tab stops.....                 | tabs(1)     |
|                                      |                        | terminate process .....                 | kill(1)     |
|                                      |                        | test — test for condition.....          | test(1)     |
|                                      |                        | test for condition .....                | test(1)     |
|                                      |                        | text editor .....                       | ex(1)       |
| display-oriented interactive         |                        | text editor .....                       | vi(1)       |
|                                      | line-oriented          | text editor .....                       | ed(1)       |
|                                      | red — line-oriented    | text editor .....                       | ed(1)       |
|                                      | split a                | text file, according to criteria.....   | csplit(1)   |
| bdiff — compare two                  |                        | text files and show differences .....   | diff(1)     |
|                                      | compare two            | text files and show differences .....   | diff(1)     |
| diffh — compare two                  |                        | text files and show differences .....   | diff(1)     |
|                                      | compare three          | text files .....                        | diff3(1)    |
| concatenate and display              |                        | text files .....                        | cat(1)      |
|                                      | simple                 | text formatter .....                    | fmt(1)      |
|                                      | display                | text in large font.....                 | banner(1)   |
|                                      | join two sorted,       | textual relational databases.....       | join(1)     |
|                                      |                        | three-way file merge.....               | merge(1)    |
| set and display date and             |                        | time .....                              | date(1)     |
|                                      | setting local          | time zone .....                         | timezone(3) |
| execution for a specified            |                        | time/suspend .....                      | sleep(1)    |
|                                      | zone/                  | timezone — setting local time.....      | timezone(3) |
| files to byte stream files/          |                        | tobyte — convert MPE record.....        | tobyte(1)   |
|                                      | date/                  | touch — change file modification .....  | touch(1)    |
|                                      | characteristics/       | tput — change terminal.....             | tput(1)     |

|                                |                                         |               |
|--------------------------------|-----------------------------------------|---------------|
|                                | tr — translation filter .....           | tr(1)         |
| create a                       | tracked alias .....                     | hash(1)       |
| language/data                  | transformation, report generation ..... | awk(1)        |
|                                | translation filter.....                 | tr(1)         |
| encode a file for safe         | transmission .....                      | uuencode(1)   |
| decode                         | transmitted binary file .....           | uudecode(1)   |
| conditions and interrupts/     | trap — intercept abnormal .....         | trap(1)       |
| find files within file         | tree .....                              | find(1)       |
|                                | true — do nothing, successfully .....   | true(1)       |
|                                | tmail — mail delivery program .....     | tmail(3)      |
|                                | tty — display terminal name .....       | tty(1)        |
| name/                          | type — tell how shell interprets .....  | type(1)       |
| determine file                 | type .....                              | file(1)       |
| values to variables/           | typeset — assign attributes and .....   | typeset(1)    |
| creation mask/                 | umask — get or set the file mode .....  | umask(1)      |
| definitions/                   | unalias — remove alias .....            | unalias(1)    |
| information/                   | uname — display system name .....       | uname(1)      |
| compression of a file/         | uncompress — Undo Lempel-Ziv .....      | uncompress(1) |
|                                | uncompress and display data .....       | zcat(1)       |
| check in a file                | under RCS .....                         | ci(1)         |
| check out a file               | under RCS .....                         | co(1)         |
| file/                          | Undo Lempel-Ziv compression of a.....   | uncompress(1) |
| tabs/                          | unexpand — compress spaces into.....    | unexpand(1)   |
| sorted file/                   | uniq — display unique lines of.....     | uniq(1)       |
| display                        | unique lines of sorted file.....        | uniq(1)       |
| files/                         | unpack — decode Huffman packed .....    | unpack(1)     |
| function/                      | unset — remove shell variable or.....   | unset(1)      |
| archiver to copy and back      | up files .....                          | cpio(1)       |
| tape archiver to copy and back | up files/USTAR-compatible.....          | tar(1)        |
| clean                          | up working files .....                  | rcsclean(1)   |
| summarize file space           | usage .....                             | du(1)         |
| functions                      | used with lex .....                     | lex(3)        |
| display                        | user and group names .....              | id(1)         |
| display                        | user name .....                         | logname(1)    |
| write to another               | user.....                               | write(1)      |
| write to all logged in         | users .....                             | wall(1)       |
| information about current      | users/display.....                      | who(1)        |
| change file                    | using diff output.....                  | patch(1)      |
| copy and back up files/        | USTAR-compatible tape archiver to.....  | tar(1)        |
| SCCS to RCS conversion         | utility .....                           | sccs2rcs(1)   |
| sort/merge                     | utility .....                           | sort(1)       |
| binary file/                   | uudecode — decode transmitted .....     | uudecode(1)   |
| safe transmission/             | uuencode — encode a file for .....      | uuencode(1)   |
| assign attributes and          | values to variables .....               | typeset(1)    |
| mark                           | variable as readonly.....               | readonly(1)   |

|                             |                                        |             |
|-----------------------------|----------------------------------------|-------------|
| declare an integer          | variable .....                         | integer(1)  |
| remove shell                | variable or function .....             | unset(1)    |
| standard environment        | variables .....                        | environ(3)  |
| attributes and values to    | variables/assign .....                 | typeset(1)  |
| interactive text editor/    | vi — display-oriented .....            | vi(1)       |
| tags file for ex, more, and | vi/produce .....                       | ctags(1)    |
| complete/                   | wait — wait for process to .....       | wait(1)     |
|                             | wait for process to complete .....     | wait(1)     |
| users/                      | wall — write to all logged in .....    | wall(1)     |
| bytes, and characters/      | wc — count of newlines, words, .....   | wc(1)       |
| interprets command name/    | whence — tell how shell .....          | whence(1)   |
| executable command/         | which — display path name for .....    | which(1)    |
| current users/              | who — display information about .....  | who(1)      |
| find files                  | within file tree .....                 | find(1)     |
| count of newlines,          | words, bytes, and characters .....     | wc(1)       |
| change                      | working directory .....                | cd(1)       |
| display                     | working directory .....                | pwd(1)      |
| clean up                    | working files .....                    | rcsclean(1) |
|                             | write — write to another user .....    | write(1)    |
|                             | write to all logged in users .....     | wall(1)     |
|                             | write to another user .....            | write(1)    |
| command lines/              | xargs — construct and execute .....    | xargs(1)    |
|                             | yacc — parser generator language ..... | yacc(1)     |
| data/                       | zcat — uncompress and display .....    | zcat(1)     |
| setting local time          | zone .....                             | timezone(3) |



# Index

! command 1-678 to 1-678  
!filter vi command 1-655 to 1-655  
!run shell ex command 1-670 to 1-670  
" comment ex command 1-671 to 1-671  
# display lines with line numbers ex command 1-666 to 1-666  
#define directive 3-6 to 3-6  
#line 1-710 to 1-710  
#undef directive 3-6 to 3-6  
\$ end of line metacharacter 1-661 to 1-661  
\$ move to end of line vi command 1-651 to 1-651  
\$STDLIST variable 1-693 to 1-704  
% move to balancing bracket vi command 1-653 to 1-653  
%prefix 1-711 to 1-711  
& all matched characters 1-662 to 1-662  
& repeat last substitute ex command 1-671 to 1-671  
& repeat last substitute vi command 1-657 to 1-657  
& shell operator 1-522 to 1-522  
( move to previous sentence vi command 1-652 to 1-652  
) move to next sentence vi command 1-652 to 1-652  
\* repetitions of expression metacharacter 1-661 to 1-661  
+ move to start of next line vi command 1-650 to 1-650  
, reverse character search vi command 1-653 to 1-653  
. any character metacharacter 1-661 to 1-661  
. command 1-189 to 1-486  
.repeat last command vi command 1-657 to 1-657  
.exrc file name 1-647 to 1-678  
.mailrc file name 1-319 to 1-319  
.profile file name A-18 to A-18, A-2 to A-2, A-2 to A-2, A-2 to A-2, 1-517 to 1-545  
.sh\_history file name A-2 to A-2, 1-267 to 1-545  
/ path name A-10 to A-10  
/ search forwards for matching string vi command 1-653 to 1-653  
/bin/sh path name 1-667 to 1-667  
/dev/mt/0m path name 1-591 to 1-591  
/etc path name 1-265 to 1-265  
/etc/helpfile path name 1-266 to 1-266  
/etc/helpindex path name 1-266 to 1-266  
/etc/init path name 1-700 to 1-700  
/etc/italic.fnt path name 1-42 to 1-42  
/etc/magic path name 1-227 to 1-227, 2-4 to 2-4  
/etc/mailx.rc path name 1-305 to 1-333  
/etc/passwd path name 1-529 to 1-529  
/etc/profile path name A-18 to A-18, 1-520 to 1-555, 3-4 to 3-4  
/etc/small.fnt path name 1-42 to 1-42  
/etc/startup.mk path name 1-351 to 1-353  
/etc/suid\_profile path name 1-517 to 1-545  
/etc/utmp path name 1-700 to 1-701  
/etc/wtmp path name 1-700 to 1-701  
/etc/yylex.c path name 1-287 to 1-288  
/etc/yyparse.c path name 1-710 to 1-711  
/tmp path name 1-179 to 1-572, 3-4 to 3-4  
/tmp/e(\*\* path name 1-202 to 1-202  
/tmp/sh(\*\* path name 1-545 to 1-545  
/tmp/stm(\*\* path name 1-563 to 1-563  
/usr/lib/lib.b path name 1-63 to 1-63  
/usr/lib/libl.a path name 1-288 to 1-288, 3-8 to 3-8  
/usr/mail path name 1-320 to 1-326  
/usr/man path name 1-362 to 1-364  
/usr/man/man1/(\*\* .1 path name 1-266 to 1-266  
: command 1-118 to 1-118  
: invoke ex command from vi command 1-657 to 1-657  
; repeat character search vi command 1-653 to 1-653  
< shift left ex command 1-670 to 1-670  
< shift left vi command 1-655 to 1-655  
< start of word metacharacter 1-661 to 1-661  
= display line number ex command 1-671 to 1-671  
> end of word metacharacter 1-661 to 1-661  
> shift right ex command 1-670 to 1-670  
> shift right vi command 1-655 to 1-655  
? display Boolean variable ex command 1-667 to 1-667  
? search backwards for matching string vi command 1-653 to 1-653  
@ delete inserted line vi command 1-659 to 1-659  
@ execute buffer ex command 1-671 to 1-671  
@ invoke macro vi command 1-657 to 1-657  
[[ move back to start of section vi command 1-653 to 1-653  
CTRL-@ repeat last insertion vi command 1-659 to 1-659  
CTRL-B scroll back page vi command 1-649 to 1-649  
CTRL-D decrement autoindent vi command 1-658 to 1-658  
CTRL-D display lines ex command 1-671 to 1-671  
CTRL-D scroll forward half screen vi command 1-649 to 1-649  
CTRL-E scroll forward a line vi command 1-649 to 1-649  
CTRL-F scroll forward page vi command 1-649 to 1-649  
CTRL-G file information vi command 1-657 to 1-657  
CTRL-H delete last character vi command 1-658 to 1-658  
CTRL-H move left vi command 1-650 to 1-650  
CTRL-J move down one line vi command 1-650 to 1-650  
CTRL-L redraw screen vi command 1-657 to 1-657  
CTRL-M move to start of next line vi command 1-650 to 1-650  
CTRL-N move down one line vi command 1-650 to 1-650  
CTRL-P move up one line vi command 1-650 to 1-650  
CTRL-Q insert character literally vi command 1-658 to 1-658  
CTRL-T increment autoindent vi command 1-658 to 1-658  
CTRL-U scroll back half screen vi command 1-649 to 1-649  
CTRL-V insert character literally vi command 1-658 to 1-678  
CTRL-W back up a word vi command 1-659 to 1-659  
CTRL-Y scroll back a line vi command 1-649 to 1-649  
CTRL-Z stop editor vi command 1-658 to 1-658

**CTRL-^** edit alternate file vi command 1-658 to 1-658  
`^` move to start of marked line vi command 1-651 to 1-651  
``` move to mark vi command 1-651 to 1-651  
`-` move to start of previous line vi command 1-651 to 1-651
`]]` move to next section vi command 1-653 to 1-653
`^` move to first non-blank vi command 1-651 to 1-651
`^` start of line metacharacter 1-661 to 1-661
`{` move back to start of paragraph vi command 1-652 to 1-652
`|` move to column vi command 1-651 to 1-651
`|` multiple commands in ex 1-659 to 1-659
`|&` shell operator 1-523 to 1-523
`}` move to next paragraph vi command 1-652 to 1-652
`~` repeat last substitute ex command 1-671 to 1-671
`~` replacement from last substitute 1-662 to 1-662
`~` toggle case of character vi command 1-657 to 1-657
`~!` invoke system command 3-9 to 3-9
`~?` display editing commands 3-9 to 3-9
`~e` invoke text editor on message 3-9 to 3-9
`~p` display message 3-9 to 3-9
`~q` discard message 3-9 to 3-9
`~r` read file into message 3-9 to 3-9
`~v` invoke visual editor on message 3-9 to 3-9
`~w` write message to file 3-9 to 3-9
0 move to start of line vi command 1-650 to 1-650

A

A append at end of line vi command 1-656 to 1-656
a append text vi command 1-656 to 1-656
a. out file name 1-71 to 1-73
abbreviate define abbreviation ex command 1-663 to 1-663
 abbreviations 1-669 to 1-669
 abbreviations, object manipulator 1-655 to 1-655
 absolute line 1-650 to 1-650
 absolute movement, vi 1-648 to 1-650
 access list 1-109 to 1-95
 access time 1-128 to 1-300
 account directory 1-496 to 1-93
 action, awk 1-29 to 1-29
 addition, dc 1-157 to 1-157
 addresses, ex 1-659 to 1-659
 alias 1-696 to 1-8
 alias command 1-10 to 1-10
 alias deletion 1-626 to 1-626
 alias, tracked 1-262 to 1-9
 all matched characters, & 1-662 to 1-662
 allnet environment variable 1-320 to 1-333
 alternate files in ex 1-672 to 1-672
 ALTGROUP command A-11 to A-11
 ALTSEC command A-6 to A-6, 3-48 to 3-48
 ALTUSER command A-6 to A-6
 analyzer, lexical 1-287 to 1-287
 any character metacharacter, . 1-661 to 1-661
 append 1-488 to 1-488
append append text ex command 1-663 to 1-663
 append at end of line vi command, **A** 1-656 to 1-656

append command 1-211 to 1-211
 append ed command 1-197 to 1-198
 append environment variable 1-321 to 1-321
 append text ex command, **append** 1-663 to 1-663
 append text vi command, **a** 1-656 to 1-656
 append to file ex command 1-669 to 1-669
 appointments 1-76 to 1-76
 ar command 1-11 to 1-14
 archive extraction 1-416 to 1-416
 archive files 1-133 to 1-416
 archive libraries 1-11 to 1-11
 archive, shell 1-541 to 1-541
 archive, tape 1-590 to 1-590
 ARGV variable 1-23 to 1-24
args display current file list ex command 1-663 to 1-663
 arguments, command 1-705 to 1-705
 arguments, print 1-437 to 1-437
 ARGV array 1-22 to 1-24
 arithmetic 1-285 to 1-599
 array, ARGV 1-22 to 1-24
 array, ENVIRON 1-21 to 1-21
 array, SYMTAB 1-21 to 1-21
 array, yytext 3-6 to 3-6
 arrays, awk 1-19 to 1-19
 arrays, dc 1-155 to 1-162
 arrays, shell 1-539 to 1-539
 ASA 1-15 to 1-15
 asa command 1-15 to 1-16
 ascii 1-41 to 1-41, 3-2 to 3-2
 ASCII-to-EBCDIC 1-169 to 1-169
 ask environment variable 1-321 to 1-321
 askbcc environment variable 1-321 to 1-321
 askcc environment variable 1-321 to 1-321
 asksub environment variable 1-321 to 1-321
 assembler programs 1-227 to 1-227
 attributes 1-639 to 1-639
 attributes, file 1-300 to 1-88
 attributes, variable 1-621 to 1-621
 author 1-109 to 1-98
 Author keyword 1-113 to 1-271
 autoindent automatic indent vi variable 1-658 to 1-676
 automatic indent vi variable, autoindent 1-658 to 1-676
 autoprnt display current line after change vi variable 1-672 to 1-672
 autoprnt environment variable 1-307 to 1-321
 autowrite vi variable 1-666 to 1-673
 awk action 1-29 to 1-29
 awk arrays 1-19 to 1-19
 awk BEGIN 1-29 to 1-29
 awk built-in arithmetic functions 1-26 to 1-26
 awk built-in string functions 1-26 to 1-26
 awk command 1-142 to 1-41, 3-2 to 3-37
 awk comments 1-18 to 1-18
 awk END 1-29 to 1-29
 awk examples 1-32 to 1-32
 awk FILENAME file being read 1-24 to 1-24
 awk FNR number of records read from file 1-24 to 1-24
 awk functions 1-28 to 1-28

awk `getline` 1-25 to 1-25
 awk NF field in current record 1-24 to 1-24
 awk NR number of records read 1-24 to 1-24
 awk operators 1-21 to 1-21
 awk patterns 1-29 to 1-29
 awk statements 1-29 to 1-29
 awk `system` 1-28 to 1-28
 awk variables 1-19 to 1-19

B

B move to previous full word `vi` command 1-651 to 1-651
b move to previous word `vi` command 1-651 to 1-651
 back up a word `vi` command, **CTRL-W** 1-659 to 1-659
 backquoting 1-534 to 1-534
 backslash 1-661 to 1-661
 backspace 1-646 to 1-646
 backspace delete last character `vi` command 1-658 to 1-658
 backspace move left `vi` command 1-650 to 1-650
 bang environment variable 1-318 to 1-321
 banner command 1-42 to 1-43
 basename command 1-188 to 1-144
 bases, `dc` 1-160 to 1-160
`bc` command 1-167 to 1-167
`bdiff` command 1-174 to 1-180
`beautify` discard control characters `vi` variable 1-673 to 1-673
 BEGIN, awk 1-29 to 1-29
`bg` command 1-274 to 1-549
 binary files 1-107 to 1-96
 binary files, compare 1-107 to 1-107
 block size 1-397 to 1-397
 block-special 1-376 to 1-376
 branch 1-109 to 1-97, 3-25 to 3-25
 branch, default 1-109 to 1-98, 3-25 to 3-25
 branch, highest 1-488 to 1-488
`break` command 1-127 to 1-68
 break up files 1-567 to 1-567
 breaking locks 1-458 to 1-95
 brief command explanations 1-265 to 1-265
 buffer 1-654 to 1-654
 buffer size `vi` variable, `maxbuffers` 1-674 to 1-674
 buffer, hold 1-506 to 1-506
 buffer, named 1-654 to 1-654
 buffer, numbered 1-654 to 1-654
 buffer, pattern 1-506 to 1-506
 buffer, unnamed 1-654 to 1-654
 built-in arithmetic functions, awk 1-26 to 1-26
 built-in commands, special 1-540 to 1-540
 built-in string functions, awk 1-26 to 1-26
 BYE command A-18 to A-18
 byte stream files 1-126 to 1-690
 byte type 2-4 to 2-4
 bytes, count 1-694 to 1-694
 bytes, swapping 1-135 to 1-169

C

C change to end of line `vi` command 1-655 to 1-655
c change `vi` command 1-654 to 1-654
`c` command 1-284 to 1-70
 C escape sequences 1-405 to 1-405
 C preprocessor 1-175 to 1-175
 C programming 1-143 to 1-143
 C source code 1-143 to 1-143
 C source files 1-227 to 1-227
`c89` command A-17 to A-17, A-17 to A-17, 1-14 to 1-75
 calculator 1-155 to 1-155
 calendar 1-76 to 1-76
 calendar command 1-76 to 1-78
 calendar file name 1-76 to 1-77
 calendar path name 1-76 to 1-76
`callci` command A-3 to A-3, 1-79 to 1-80
 caller 1-112 to 1-112
 carriage control 1-15 to 1-15
 case sensitivity 3-25 to 3-25
 case shell command 1-523 to 1-523
`cat` command 1-132 to 1-82
 CCOMXL.PUB.SYS file name 1-72 to 1-72
cd change directory `ex` command 1-663 to 1-663
`cd` command 1-10 to 1-85, 3-15 to 3-4
 CDPATH environment variable 1-542 to 1-84
 center lines 1-238 to 1-238
change change lines `ex` command 1-663 to 1-663
 change character `vi` command, **s** 1-655 to 1-655
 change directory `ex` command, **cd** 1-663 to 1-663
 change directory `ex` command, **chdir** 1-663 to 1-663
 change `ed` command 1-197 to 1-197
 change environment variables 1-515 to 1-515
 change file `ed` command 1-198 to 1-198
 change file mode 1-88 to 1-88
 change line `vi` command, **S** 1-655 to 1-655
 change lines `ex` command, **change** 1-663 to 1-663
 change modification time 1-605 to 1-605
 change time 1-232 to 1-232
 change to end of line `vi` command, **C** 1-655 to 1-655
 change `vi` command, **c** 1-654 to 1-654
 changes, combine 1-367 to 1-367
 character-special 1-376 to 1-376
 characters, control 1-644 to 1-81
 characters, count 1-694 to 1-694
 characters, translate 1-611 to 1-611
 characters, unprintable 1-81 to 1-81
 characters, wild card 1-538 to 1-538
chdir change directory `ex` command 1-663 to 1-663
 check in 1-109 to 1-104
 check out 1-456 to 1-117
 checked-out revision 1-474 to 1-474
 checksum 1-105 to 1-582
 CHGROUP command 1-555 to 1-555
`chgrp` command 1-86 to 1-87
`chmod` command A-10 to A-10, A-11 to A-11, A-11 to A-11, A-13 to A-13, A-5 to A-5, A-8 to A-8, 1-178 to 1-91

choose command interpreter `ex` variable, `shell` 1-668 to 1-675
`chown` command A-12 to A-12, 1-87 to 1-94
`ci` command 1-100 to 1-104, 3-10 to 3-26
`CI, MPE/iX` 1-13 to 1-77
`cksum` command 1-105 to 1-106
 class regular expressions 1-662 to 1-662
 cleanup working files 1-466 to 1-466
 closing standard input 1-537 to 1-537
 closing standard output 1-537 to 1-537
`cmd` environment variable 1-312 to 1-332
`cmp` command 1-106 to 1-108
`co` command 1-104 to 1-117, 3-11 to 3-26
`co-process` 1-523 to 1-537
`colon` command 1-118 to 1-118
`columns` 1-147 to 1-69
`COLUMNS` environment variable 1-201 to 1-69, 3-3 to 3-38
 combine changes 1-367 to 1-367
`comm` command 1-108 to 1-120
 command alias 1-8 to 1-8
 command arguments 1-705 to 1-705
`command` command 1-121 to 1-122
 command construction 1-705 to 1-705
 command editing 1-224 to 1-545
 command execution 1-209 to 1-705
 command explanations, brief 1-265 to 1-265
 command format, `ex` 1-659 to 1-659
 command history 1-224 to 1-453
 command line options 1-253 to 1-253
 command mode 1-307 to 1-646
 command names 1-696 to 1-698
 command numbers 1-225 to 1-225
 command substitution 1-534 to 1-534
 command template 1-705 to 1-705
`command`, `!` 1-678 to 1-678
`command`, `.` 1-189 to 1-486
`command`, `:` 1-118 to 1-118
`command`, `alias` 1-10 to 1-10
`command`, `ALTGROUP` A-11 to A-11
`command`, `ALTSEC` A-6 to A-6, 3-48 to 3-48
`command`, `ALTUSER` A-6 to A-6
`command`, `append` 1-211 to 1-211
`command`, `ar` 1-11 to 1-14
`command`, `asa` 1-15 to 1-16
`command`, `awk` 1-142 to 1-41, 3-2 to 3-37
`command`, `banner` 1-42 to 1-43
`command`, `basename` 1-188 to 1-44
`command`, `bc` 1-167 to 1-67
`command`, `bdiff` 1-174 to 1-180
`command`, `bg` 1-274 to 1-549
`command`, `break` 1-127 to 1-68
`command`, `BYE` A-18 to A-18
`command`, `c` 1-284 to 1-70
`command`, `c89` A-17 to A-17, A-17 to A-17, 1-14 to 1-75
`command`, `calendar` 1-76 to 1-78
`command`, `callci` A-3 to A-3, 1-79 to 1-80
`command`, `cat` 1-132 to 1-82
`command`, `cd` 1-10 to 1-85, 3-15 to 3-4
`command`, `CHGROUP` 1-555 to 1-555
`command`, `chgrp` 1-86 to 1-87
`command`, `chmod` A-10 to A-10, A-11 to A-11, A-11 to A-11, A-13 to A-13, A-5 to A-5, A-8 to A-8, 1-178 to 1-91
`command`, `chown` A-12 to A-12, 1-87 to 1-94
`command`, `ci` 1-100 to 1-104, 3-10 to 3-26
`command`, `cksum` 1-105 to 1-106
`command`, `cmp` 1-106 to 1-108
`command`, `co` 1-104 to 1-117, 3-11 to 3-26
`command`, `colon` 1-118 to 1-118
`command`, `comm` 1-108 to 1-120
`command`, `command` 1-121 to 1-122
`command`, `compress` 1-123 to 1-126, 2-10 to 2-3
`command`, `continue` 1-127 to 1-127
`command`, `COPY` 1-130 to 1-79
`command`, `cp` 1-128 to 1-132
`command`, `cpio` 1-126 to 1-137, 2-10 to 2-6
`command`, `create` 1-375 to 1-624
`command`, `crypt` 1-206 to 1-689
`command`, `csplit` 1-138 to 1-142, 3-36 to 3-36
`command`, `ctags` 1-143 to 1-146, 2-7 to 2-7
`command`, `cut` 1-147 to 1-149
`command`, `date` 1-150 to 1-154, 3-53 to 3-53
`command`, `dc` 1-155 to 1-167
`command`, `dd` 1-137 to 1-173
`command`, `diff` 1-106 to 1-180, 3-15 to 3-15
`command`, `diff3` 1-180 to 1-184
`command`, `diffb` 1-185 to 1-186
`command`, `diffh` 1-174 to 1-180
`command`, `dir` 1-283 to 1-283
`command`, `dirname` 1-187 to 1-188
`command`, `dot` 1-520 to 1-190
`command`, `du` 1-191 to 1-192
`command`, `echo` 1-193 to 1-194, 3-3 to 3-3
`command`, `ed` 1-174 to 1-206, 3-32 to 3-9
`command`, `egrep` 1-259 to 1-41, 3-36 to 3-36
`command`, `emacs` 1-516 to 1-516
`command`, `env` 1-207 to 1-208, 3-4 to 3-4
`command`, `eval` 1-209 to 1-210
`command`, `ex` 1-143 to 1-211, 2-7 to 2-7, 3-32 to 3-32
`command`, `exec` 1-210 to 1-213
`command`, `exit` 1-214 to 1-214
`command`, `expand` 1-215 to 1-216
`command`, `export` 1-217 to 1-218, 3-3 to 3-3
`command`, `expr` 1-219 to 1-222, 3-32 to 3-37
`command`, `exrecover` 1-666 to 1-666
`command`, `false` 1-223 to 1-223
`command`, `fc` 1-10 to 1-226, 3-38 to 3-45
`command`, `FCOPY` 1-243 to 1-243
`command`, `fg` 1-274 to 1-658
`command`, `fgrep` 1-259 to 1-535
`command`, `file` 1-227 to 1-230, 2-4 to 2-5
`command`, `find` 1-133 to 1-237
`command`, `fmt` 1-238 to 1-240
`command`, `fold` 1-241 to 1-242
`command`, `frombyte` A-3 to A-3, 1-243 to 1-244
`command`, `functions` 1-10 to 1-246
`command`, `getconf` 1-154 to 1-252
`command`, `getopt` 1-253 to 1-255

Index-4

command, getopts 1-255 to 1-258
 command, glob 3-32 to 3-32
 command, grep 1-206 to 1-261, 3-32 to 3-37
 command, gres 1-261 to 1-261
 command, hash 1-10 to 1-262
 command, head 1-263 to 1-264
 command, help 1-265 to 1-266
 command, history 1-10 to 1-268
 command, id 1-269 to 1-270
 command, ident 1-104 to 1-272, 3-26 to 3-26
 command, insert 1-211 to 1-211
 command, integer 1-10 to 1-273
 command, intro 1-1 to 1-1
 command, jobs 1-274 to 1-275
 command, join 1-276 to 1-279
 command, kill A-16 to A-16, A-16 to A-16, 1-274 to 1-282
 command, lc 1-121 to 1-284
 command, let 1-10 to 1-286
 command, lex 1-227 to 1-291, 3-6 to 3-8
 command, line 1-292 to 1-292
 command, linkedit 1-74 to 1-74
 command, LISTFILE A-17 to A-17, A-17 to A-17
 command, LISTUSER 1-77 to 1-77
 command, ln 1-293 to 1-297
 command, locale 1-297 to 1-297
 command, login 1-320 to 1-320, 3-3 to 3-3
 command, logname 1-270 to 1-298
 command, ls A-13 to A-13, 1-106 to 1-303
 command, mail 1-325 to 1-465
 command, mailx 1-240 to 1-333, 3-3 to 3-55
 command, make 1-14 to 1-361
 command, man 1-266 to 1-366
 command, merge 1-367 to 1-369
 command, mesg 1-370 to 1-371
 command, mkdir A-12 to A-12, 1-372 to 1-373
 command, mkfifo 1-374 to 1-375
 command, mknod A-19 to A-19, 1-375 to 1-377
 command, more 1-143 to 1-385, 2-7 to 2-7
 command, mv 1-132 to 1-390
 command, NEWACCT A-11 to A-11, A-11 to A-11
 command, NEWUSER A-6 to A-6
 command, nice 1-391 to 1-392
 command, nl 1-393 to 1-396
 command, nohup 1-392 to 1-392
 command, od 1-397 to 1-401
 command, pack 1-126 to 1-404
 command, paste 1-149 to 1-407
 command, patch 1-180 to 1-413
 command, pathchk 1-187 to 1-415
 command, pax A-19 to A-19, 1-126 to 1-431, 2-3 to 2-6, 3-5 to 3-5
 command, pcat 1-363 to 1-432
 command, pr 1-216 to 1-436
 command, preserve 1-667 to 1-667
 command, print 1-226 to 1-438
 command, printenv 1-208 to 1-208
 command, printf 1-194 to 1-441, 2-4 to 2-4
 command, ps 1-281 to 1-450
 command, pwd 1-10 to 1-452
 command, r 1-10 to 1-455
 command, rcs 1-102 to 1-465, 3-10 to 3-26
 command, RCS 1-460 to 1-460
 command, rcsclean 1-104 to 1-469, 3-26 to 3-26
 command, rcsdiff 1-104 to 1-473, 3-26 to 3-26
 command, rcsmerge 1-104 to 1-477, 3-26 to 3-26
 command, read 1-226 to 1-480
 command, readonly 1-481 to 1-482
 command, red 1-195 to 1-206
 command, renice 1-392 to 1-485
 command, return 1-214 to 1-486
 command, rev 1-487 to 1-487
 command, rlog 1-104 to 1-493, 3-26 to 3-26
 command, rm 1-132 to 1-496
 command, rmdir 1-373 to 1-498, 3-49 to 3-49
 command, rsh 1-519 to 1-521
 command, rvi 1-675 to 1-675
 command, sccs2rcs 1-499 to 1-505
 command, sed 1-142 to 1-514, 3-32 to 3-37
 command, set 1-10 to 1-518, 3-38 to 3-45
 command, SETMSG 1-371 to 1-371
 command, sh A-19 to A-19, 1-10 to 1-555, 3-28 to 3-45
 command, shell 1-678 to 1-678
 command, shift 1-555 to 1-556
 command, SHOWJOB 1-701 to 1-701
 command, sleep 1-557 to 1-558
 command, sort 1-120 to 1-566, 2-7 to 2-7, 3-2 to 3-2
 command, SORT 1-566 to 1-566
 command, sortgen 1-566 to 1-566
 command, split 1-139 to 1-569
 command, strings 1-570 to 1-571
 command, strip 1-572 to 1-574
 command, stty A-19 to A-19, A-19 to A-19, 1-575 to 1-581
 command, sum 1-105 to 1-583
 command, switch 1-523 to 1-523
 command, tabs 1-584 to 1-587
 command, tail 1-264 to 1-589
 command, talk 1-370 to 1-704
 command, tar A-19 to A-19, 1-126 to 1-595, 2-10 to 2-9
 command, tee 1-596 to 1-597
 command, TELL 1-371 to 1-704
 command, test 1-222 to 1-602
 command, tobyte A-3 to A-3, 1-244 to 1-604
 command, touch 1-154 to 1-607
 command, tput 1-608 to 1-610
 command, tr 1-173 to 1-614
 command, trap 1-189 to 1-616
 command, true 1-118 to 1-617
 command, tsmail 1-333 to 1-333, 3-54 to 3-55
 command, tty A-19 to A-19, A-19 to A-19, 1-618 to 1-619
 command, type 1-550 to 1-620
 command, typeset 1-10 to 1-623
 command, ulimit 1-555 to 1-555
 command, umask 1-373 to 1-625
 command, unalias 1-10 to 1-626
 command, uname 1-149 to 1-628

command, uncompress 1-103 to 1-631
 command, unexpand 1-216 to 1-633
 command, uniq 1-108 to 1-636
 command, unpack 1-126 to 1-638
 command, unset 1-555 to 1-640
 command, uuencode 1-641 to 1-643
 command, uuencode 1-641 to 1-645
 command, vi 1-143 to 1-690, 2-7 to 2-7, 3-32 to 3-9, 7
 command, wait 1-275 to 1-691
 command, wall 1-371 to 1-693
 command, wc 1-106 to 1-695
 command, whence 1-555 to 1-697
 command, which 1-696 to 1-699
 command, who 1-443 to 1-701
 command, wrapmargin 1-678 to 1-678
 command, write 1-370 to 1-704
 command, xargs 1-705 to 1-709
 command, yacc 1-227 to 1-714, 3-8 to 3-8
 command, z 1-677 to 1-677
 command, zcat 1-126 to 1-716
 commands vi, scrolling 1-649 to 1-649
 commands, scrolling 1-648 to 1-648
 commands, shell 1-521 to 1-521
 commands, shell reserved word 1-523 to 1-523
 commands, special built-in 1-540 to 1-540
 commands, vi manipulation 1-648 to 1-648
 commands, vi movement 1-648 to 1-648
 commands, vi text insertion 1-648 to 1-648
 comment ex command, " 1-671 to 1-671
 comment leader 1-456 to 1-456
 comments, awk 1-18 to 1-18
 comments, multi-line 1-456 to 1-456
 comments, shell 1-521 to 1-521
 compare binary files 1-107 to 1-107
 compare RCS revision 1-470 to 1-470
 compare sorted files 1-119 to 1-119
 compare three files 1-181 to 1-181
 compare two files 1-174 to 1-185
 compiler, MPE/iX 1-75 to 1-75
 component directory pax archive 1-416 to 1-416
 component file pax archive 1-416 to 1-416
 compress command 1-123 to 1-126, 2-10 to 2-3
 compressing files 1-402 to 1-637
 compression 1-109 to 1-95
 concatenate files 1-81 to 1-81
 concatenate lines 1-405 to 1-405
 condition, test 1-598 to 1-598
 conditional expression, test 1-598 to 1-598
 conf.c path name 1-376 to 1-376
 configuration file, RCS 1-458 to 1-458
 configuration options, ignore 1-467 to 1-467
 configuration variable 1-247 to 1-247
 configuration, machine 1-627 to 1-627
 conflicting path name 1-293 to 1-293
 considerations, implementation A-1 to A-19
 context, previous 1-651 to 1-651
 context-dependent movement, vi 1-648 to 1-651
 continue command 1-127 to 1-127
 control characters 1-644 to 1-81

control, carriage 1-15 to 1-15
 conv environment variable 1-305 to 1-333
 conversion buffer 1-171 to 1-171
 conversion, fixed to variable record 1-170 to 1-170
 conversion, lowercase 1-169 to 1-169
 conversion, SCCS 1-499 to 1-499
 conversion, uppercase 1-170 to 1-170
 conversion, variable to fixed record 1-169 to 1-169
 convert spaces to tabs 1-632 to 1-632
 CONVFMT variable 1-19 to 1-19
 COPY command 1-130 to 1-79
copy copy lines ex command 1-663 to 1-663
 copy ed command 1-200 to 1-200
 copy lines 1-292 to 1-292
 copy lines ex command, **copy** 1-663 to 1-663
 copy text ex command, **t** 1-668 to 1-668
 copying data with format conversion 1-168 to 1-168
 copying files 1-128 to 1-128
 copying selectively 1-147 to 1-147
 count (vi) 1-649 to 1-649
 count bytes 1-694 to 1-694
 count characters 1-694 to 1-694
 count lines 1-694 to 1-694
 count words 1-694 to 1-694
 cp command 1-128 to 1-132
 cpio command 1-126 to 1-137, 2-10 to 2-6
 cpio file format 1-137 to 1-595, 2-10 to 2-3
 create command 1-375 to 1-624
 create directory 1-372 to 1-372
 create RCS file 1-457 to 1-95
 crt environment variable 1-313 to 1-322
 crypt command 1-206 to 1-689
 csplit command 1-138 to 1-142, 3-36 to 3-36
 ctags command 1-143 to 1-146, 2-7 to 2-7
 ctags file format 1-687 to 1-687
 CTRL 1-646 to 1-646
CTRL-] tag vi command 1-653 to 1-653
 current directory 1-457 to 1-457
 current mail message 1-307 to 1-307
 current position pointer 1-648 to 1-648
 current working directory 1-451 to 1-83
 cursor 1-648 to 1-648
 cut command 1-147 to 1-149
 Cyclical Redundancy Check 1-105 to 1-582

D

D delete to end of line vi command 1-655 to 1-655
d delete vi command 1-655 to 1-655
 databases, join 1-276 to 1-276
 date 1-109 to 1-110
 date command 1-150 to 1-154, 3-53 to 3-53
 date format 1-110 to 1-110
 Date keyword 1-113 to 1-271
 date range 1-488 to 1-488
 dc addition 1-157 to 1-157
 dc arrays 1-155 to 1-162
 dc bases 1-160 to 1-160
 dc command 1-155 to 1-167

dc division 1-158 to 1-158
 dc executing strings 1-161 to 1-161
 dc exponentiation 1-158 to 1-158
 dc modulo 1-158 to 1-158
 dc multiplication 1-158 to 1-158
 dc registers 1-155 to 1-155
 dc remainder 1-158 to 1-158
 dc scaling factor 1-157 to 1-157
 dc square root 1-159 to 1-159
 dc subtraction 1-158 to 1-158
 dd command 1-137 to 1-173
 DEAD environment variable 1-316 to 1-325
 dead.letter file name 1-322 to 1-326
 decoding 1-641 to 1-644
 decrement autoindent vi command, **CTRL-D** 1-658 to 1-658
 default branch 1-109 to 1-98, 3-25 to 3-25
 define abbreviation ex command, **abbreviate** 1-663 to 1-663
 define macro vi command, **map** 1-665 to 1-665
 delay execution 1-557 to 1-557
 delete 1-113 to 1-488
 delete abbreviation ex command, **unabbreviate** 1-669 to 1-669
 delete character vi command, **x** 1-655 to 1-655
 delete ed command 1-197 to 1-197
delete ex command 1-664 to 1-664
 delete inserted line vi command, **@** 1-659 to 1-659
 delete last character vi command, **backspace** 1-658 to 1-658
 delete last character vi command, **CTRL-H** 1-658 to 1-658
 delete macro ex command, **unmap** 1-669 to 1-669
 delete previous character vi command, **X** 1-655 to 1-655
 delete to end of line vi command, **D** 1-655 to 1-655
 delete vi command, **d** 1-655 to 1-655
 deletion, alias 1-626 to 1-626
 delta 3-24 to 3-25
 descriptive text 1-456 to 1-98
 descriptor, file 1-536 to 1-599
 desk calculator 1-155 to 1-155
 device type 1-376 to 1-376
 DFA table entries, number of 1-287 to 1-287
 DFA table, **lex** 1-287 to 1-287
 diagnostic output, suppress 1-467 to 1-470
 dictionary sort 1-559 to 1-559
diff command 1-106 to 1-180, 3-15 to 3-15
diff3 command 1-180 to 1-184
diffb command 1-185 to 1-186
 differences, minimal 1-175 to 1-175
diffh command 1-174 to 1-180
dir command 1-283 to 1-283
 directive, **#define** 3-6 to 3-6
 directive, **#undef** 3-6 to 3-6
 directory 1-113 to 1-99
 directory pax archive, component 1-416 to 1-416
 directory substitution 1-529 to 1-529
 directory temporary directory vi variable 1-673 to 1-673
 directory, account 1-496 to 1-93
 directory, create 1-372 to 1-372
 directory, current 1-457 to 1-457
 directory, group 1-496 to 1-93
 directory, home 1-320 to 1-83
 directory, list 1-283 to 1-299
 directory, previous 1-544 to 1-544
 directory, remove 1-497 to 1-497
 directory, root 1-496 to 1-93
 directory, search 1-231 to 1-231
 directory, working 1-451 to 1-83
 dirname command 1-187 to 1-188
 discard control characters vi variable, **beautify** 1-673 to 1-673
 discard message, **~q** 3-9 to 3-9
 disk space 1-191 to 1-191
 disk usage 1-191 to 1-191
 display Boolean variable ex command, **?** 1-667 to 1-667
 display current file list ex command, **args** 1-663 to 1-663
 display current line after change vi variable, **autoprint** 1-672 to 1-672
 display editing commands, **~?** 3-9 to 3-9
 display environment variables 1-515 to 1-515
 display files 1-433 to 1-81
 display line number ed command 1-201 to 1-201
 display line number ex command, **=** 1-671 to 1-671
 display lines ex command, **print** 1-666 to 1-666
 display lines ex command, **z** 1-670 to 1-670
 display lines ex command, **CTRL-D** 1-671 to 1-671
 display lines in verbose form ex command, **list** 1-665 to 1-665
 display lines in verbose form vi variable, **list** 1-674 to 1-674
 display lines with line numbers ex command, **#** 1-666 to 1-666
 display lines with line numbers ex command, **number** 1-666 to 1-666
 display lines with line numbers vi variable, **number** 1-674 to 1-674
 display message, **~p** 3-9 to 3-9
 display terminal name 1-618 to 1-618
 display variables 1-621 to 1-621
 displaying login information 1-700 to 1-700
 displaying strings 1-570 to 1-570
 distance to scroll vi variable, **scroll** 1-670 to 1-675
 distribution, software 1-97 to 1-97
 division, dc 1-158 to 1-158
 do ed like substitutions vi variable, **edcompatible** 1-673 to 1-673
 do not write on file vi variable, **readonly** 1-647 to 1-675
 do not write to existing file vi variable, **writeany** 1-677 to 1-677
 document, here 1-537 to 1-537
 documentation, online 1-362 to 1-362
dot command 1-520 to 1-190
dot environment variable 1-322 to 1-322
 double-spaced output 1-433 to 1-433
du command 1-191 to 1-192
 dumping files 1-397 to 1-397
 duplicate lines, removing 1-634 to 1-634

duplicate output stream 1-596 to 1-596

E

E move to end of full word `vi` command 1-651 to 1-651

e move to end of word `vi` command 1-652 to 1-652

EBCDIC-to-ASCII 1-169 to 1-169

`echo` command 1-193 to 1-194, 3-3 to 3-3

ECHO environment variable 1-14 to 1-75

ECHO `lex` statement 3-8 to 3-8

`ed` command 1-174 to 1-206, 3-32 to 3-9

`ed` command, `append` 1-197 to 1-198

`ed` command, `change` 1-197 to 1-197

`ed` command, `change file` 1-198 to 1-198

`ed` command, `copy` 1-200 to 1-200

`ed` command, `delete` 1-197 to 1-197

`ed` command, `display line number` 1-201 to 1-201

`ed` command, `edit file` 1-197 to 1-197

`ed` command, `encrypt` 1-201 to 1-201

`ed` command, `global` 1-198 to 1-198

`ed` command, `insert` 1-198 to 1-198

`ed` command, `join` 1-198 to 1-198

`ed` command, `list lines` 1-199 to 1-199

`ed` command, `mark` 1-199 to 1-199

`ed` command, `move` 1-199 to 1-199

`ed` command, `number lines` 1-199 to 1-199

`ed` command, `print lines` 1-199 to 1-199

`ed` command, `prompt` 1-199 to 1-199

`ed` command, `quiet mode` 1-198 to 1-198

`ed` command, `quit` 1-199 to 1-199

`ed` command, `read file` 1-199 to 1-199

`ed` command, `shell escape` 1-201 to 1-201

`ed` command, `substitute` 1-199 to 1-199

`ed` command, `undo change` 1-200 to 1-200

`ed` command, `write file` 1-201 to 1-201

`ed` prompt string 1-195 to 1-195

`ed.hup` file name 1-202 to 1-202

`edcompatible` `do` `ed` like substitutions `vi` variable 1-673 to 1-673

`edit` alternate file `vi` command, `CTRL-^` 1-658 to 1-658

`edit` edit new file `ex` command 1-664 to 1-669

`edit` file `ed` command 1-197 to 1-197

`edit` new file `ex` command, `edit` 1-664 to 1-669

`edit` new file `ex` command, `ex` 1-664 to 1-664

`edit` next file `ex` command, `next` 1-666 to 1-666

`edit` script 1-175 to 1-506

EDITOR environment variable 1-310 to 1-545, 3-10 to 3-9

editor version `ex` command, `version` 1-669 to 1-669

editor, stream 1-506 to 1-506

editor, text 1-174 to 1-506

`egrep` command 1-259 to 1-41, 3-36 to 3-36

electronic mail 1-304 to 1-304

`elif` shell command 1-524 to 1-524

`else` shell command 1-524 to 1-524

EMACS 1-542 to 1-542

`emacs` command 1-516 to 1-516

encoding 1-641 to 1-644

encoding files 1-402 to 1-637

encoding, Huffman 1-402 to 1-637

`encrypt` `ed` command 1-201 to 1-201

end of file 1-588 to 1-588

end of line metacharacter, `$` 1-661 to 1-661

end of word metacharacter, `>` 1-661 to 1-661

END, `awk` 1-29 to 1-29

`enter` `ex` mode `vi` command, `Q` 1-657 to 1-657

`enter` open mode `ex` command, `open` 1-666 to 1-666

`enter` `vi` mode `ex` command, `visual` 1-669 to 1-669

entries, number of DFA table 1-287 to 1-287

entries, number of move table 1-287 to 1-287

entries, number of NFA table 1-287 to 1-287

`env` command 1-207 to 1-208, 3-4 to 3-4

ENV environment variable 1-10 to 1-678, 3-3 to 3-3

`environ` 1-202 to 1-677, 2-5 to 2-5, 3-3 to 3-4

ENVIRON array 1-21 to 1-21

environment 1-207 to 1-207

environment variable 1-515 to 1-515

environment variable, `allnet` 1-320 to 1-333

environment variable, `append` 1-321 to 1-321

environment variable, `ask` 1-321 to 1-321

environment variable, `askbcc` 1-321 to 1-321

environment variable, `askcc` 1-321 to 1-321

environment variable, `asksub` 1-321 to 1-321

environment variable, `autoprint` 1-307 to 1-321

environment variable, `bang` 1-318 to 1-321

environment variable, `CDPATH` 1-542 to 1-84

environment variable, `cmd` 1-312 to 1-332

environment variable, `COLUMNS` 1-201 to 1-69, 3-3 to 3-38

environment variable, `conv` 1-305 to 1-333

environment variable, `crt` 1-313 to 1-322

environment variable, `DEAD` 1-316 to 1-325

environment variable, `dot` 1-322 to 1-322

environment variable, `ECHO` 1-14 to 1-75

environment variable, `EDITOR` 1-310 to 1-545, 3-10 to 3-9

environment variable, `ENV` 1-10 to 1-678, 3-3 to 3-3

environment variable, `ERRNO` 1-542 to 1-554

environment variable, `escape` 1-316 to 1-332

environment variable, `EXINIT` 1-647 to 1-678

environment variable, `FCEDIT` 1-225 to 1-542

environment variable, `flipr` 1-313 to 1-324

environment variable, `folder` 1-309 to 1-330

environment variable, `header` 1-323 to 1-323

environment variable, `HELP` 1-265 to 1-265

environment variable, `HELPCMD` 1-265 to 1-265

environment variable, `HISTFILE` 1-224 to 1-548

environment variable, `HISTSIZE` 1-224 to 1-601

environment variable, `hold` 1-307 to 1-323

environment variable, `HOME` A-2 to A-2, A-2 to A-2, 1-201

to 1-84, 3-3 to 3-3

environment variable, `IFS` 1-478 to 1-543

environment variable, `ignore` 1-305 to 1-323

environment variable, `ignoreeof` 1-322 to 1-323

environment variable, `indent` 1-323 to 1-323

environment variable, `indentprefix` 1-317 to 1-323

environment variable, `keep` 1-323 to 1-323

environment variable, `keepsave` 1-306 to 1-323

environment variable, `LC_COLLATE` 1-249 to 1-612, 3-30

to 3-30

environment variable, `LC_CTYPE` 1-612 to 1-613, 3-30 to

3-30

environment variable, LC_MESSAGES 1-233 to 1-494
environment variable, LC_TIME 1-152 to 1-152
environment variable, LINE 1-384 to 1-384
environment variable, LINENO 1-543 to 1-554
environment variable, LINES 1-382 to 1-678, 3-3 to 3-3
environment variable, LISTER 1-323 to 1-323
environment variable, LOGNAME 1-298 to 1-320, 3-3 to 3-3
environment variable, MAIL 1-543 to 1-543
environment variable, MAILCHECK 1-543 to 1-601
environment variable, MAILDIR 1-320 to 1-326
environment variable, MAILER 1-458 to 1-77, 3-3 to 3-3
environment variable, MAILPATH 1-527 to 1-543
environment variable, MAILRC 1-319 to 1-332
environment variable, MAKEFLAGS 1-350 to 1-355
environment variable, MAKESTARTUP 1-335 to 1-353
environment variable, MANPAGER 1-363 to 1-365
environment variable, MANPATH 1-362 to 1-365
environment variable, MBOX 1-323 to 1-326
environment variable, metoo 1-324 to 1-324
environment variable, MORE 1-382 to 1-382
environment variable, noallnet 1-321 to 1-321
environment variable, noappend 1-321 to 1-321
environment variable, noask 1-321 to 1-321
environment variable, noaskbcc 1-321 to 1-321
environment variable, noaskcc 1-321 to 1-321
environment variable, noasksub 1-321 to 1-321
environment variable, noautoprint 1-320 to 1-321
environment variable, nobang 1-321 to 1-321
environment variable, nocrt 1-322 to 1-322
environment variable, nodot 1-322 to 1-322
environment variable, noflipr 1-322 to 1-322
environment variable, nohold 1-323 to 1-323
environment variable, noignore 1-323 to 1-323
environment variable, noignoreeof 1-323 to 1-323
environment variable, nokeep 1-323 to 1-323
environment variable, nokeepsave 1-323 to 1-323
environment variable, nometoo 1-324 to 1-324
environment variable, nooutfolder 1-324 to 1-324
environment variable, nopage 1-324 to 1-324
environment variable, noquiet 1-324 to 1-324
environment variable, norecord 1-324 to 1-324
environment variable, nosave 1-325 to 1-325
environment variable, nosendwait 1-325 to 1-325
environment variable, noshowto 1-325 to 1-325
environment variable, noSign 1-325 to 1-325
environment variable, nosign 1-325 to 1-325
environment variable, OLDPWD 1-529 to 1-83
environment variable, onehop 1-324 to 1-333
environment variable, OPTIND 1-546 to 1-601
environment variable, out 1-537 to 1-537
environment variable, outfolder 1-324 to 1-324
environment variable, page 1-312 to 1-324
environment variable, PAGER 1-313 to 1-364
environment variable, PATH 1-121 to 1-9, 3-3 to 3-3
environment variable, PID 1-544 to 1-544
environment variable, prompt 1-324 to 1-324
environment variable, PS1 1-527 to 1-544
environment variable, PS2 1-479 to 1-544
environment variable, PS3 1-525 to 1-554
environment variable, PS4 1-527 to 1-544
environment variable, PWD 1-516 to 1-83
environment variable, quiet 1-324 to 1-324
environment variable, RANDOM 1-544 to 1-544
environment variable, RCSPATH 1-103 to 1-501, 3-12 to 3-15
environment variable, record 1-304 to 1-324
environment variable, REPLY 1-478 to 1-544
environment variable, Replyall 1-322 to 1-324
environment variable, replyall 1-333 to 1-333
environment variable, save 1-325 to 1-325
environment variable, screen 1-311 to 1-325
environment variable, SECONDS 1-544 to 1-544
environment variable, sendmail 1-325 to 1-333
environment variable, sendwait 1-325 to 1-333
environment variable, SHELL 1-197 to 1-678, 3-4 to 3-4
environment variable, SHELLMETAS 1-348 to 1-348
environment variable, showto 1-325 to 1-325
environment variable, Sign 1-316 to 1-325
environment variable, sign 1-316 to 1-325
environment variable, TERM 1-325 to 1-678, 3-4 to 3-4
environment variable, TERMINFO 1-201 to 1-679
environment variable, TMOUT 1-544 to 1-601
environment variable, TMP 1-678 to 1-679
environment variable, TMPDIR 1-126 to 1-73, 3-14 to 3-4
environment variable, toplines 1-314 to 1-325
environment variable, TZ 1-153 to 1-606, 3-4 to 3-53
environment variable, VAR 1-118 to 1-118
environment variable, VISUAL 1-315 to 1-545, 3-10 to 3-9
environment variables, change 1-515 to 1-515
environment variables, display 1-515 to 1-515
environment variables, exporting 1-217 to 1-217
environment, shell execution 1-539 to 1-539
environment, subshell 1-539 to 1-539
ERRNO environment variable 1-542 to 1-554
errorbells precedes error message with alert vi variable 1-673 to 1-673
ESC leave insert mode vi command 1-659 to 1-659
escape environment variable 1-316 to 1-332
escape sequences 1-528 to 1-528
escape sequences, C 1-405 to 1-405
eval command 1-209 to 1-210
ex addresses 1-659 to 1-659
ex as filter 1-647 to 1-647
ex command 1-143 to 1-211, 2-7 to 2-7, 3-32 to 3-32
ex command format 1-659 to 1-659
ex command, ! run shell 1-670 to 1-670
ex command, " comment 1-671 to 1-671
ex command, # display lines with line numbers 1-666 to 1-666
ex command, & repeat last substitute 1-671 to 1-671
ex command, < shift left 1-670 to 1-670
ex command, = display line number 1-671 to 1-671
ex command, > shift right 1-670 to 1-670
ex command, ? display Boolean variable 1-667 to 1-667
ex command, @ execute buffer 1-671 to 1-671
ex command, abbreviate define abbreviation 1-663 to 1-663
ex command, append append text 1-663 to 1-663

ex command, **append** to file 1-669 to 1-669
 ex command, **args** display current file list 1-663 to 1-663
 ex command, **cd** change directory 1-663 to 1-663
 ex command, **change** change lines 1-663 to 1-663
 ex command, **chdir** change directory 1-663 to 1-663
 ex command, **copy** copy lines 1-663 to 1-663
 ex command, **delete** 1-664 to 1-664
 ex command, **edit** edit new file 1-664 to 1-669
 ex command, **ex** edit new file 1-664 to 1-664
 ex command, **file** obtain file information 1-664 to 1-664
 ex command, **g** global find 1-664 to 1-664
 ex command, **insert** insert text 1-665 to 1-665
 ex command, **join** join lines 1-665 to 1-665
 ex command, **k** mark text 1-665 to 1-665
 ex command, **list** display lines in verbose form 1-665 to 1-665
 ex command, **mark** mark text 1-665 to 1-665
 ex command, **move** move text 1-665 to 1-665
 ex command, **next** edit next file 1-666 to 1-666
 ex command, **number** display lines with line numbers 1-666 to 1-666
 ex command, **open** enter open mode 1-666 to 1-666
 ex command, **pipe** 1-669 to 1-669
 ex command, **preserve** saves buffer for recovery 1-666 to 1-666
 ex command, **print** display lines 1-666 to 1-666
 ex command, **put** paste lines 1-666 to 1-666
 ex command, **quit** quit editing session 1-666 to 1-666
 ex command, **read** read file 1-667 to 1-667
 ex command, **recover** recover file 1-667 to 1-667
 ex command, **rewind** restart file list 1-667 to 1-667
 ex command, **set** set variable value 1-672 to 1-672
 ex command, **shell** invoke subshell 1-668 to 1-668
 ex command, **source** run command file 1-668 to 1-678
 ex command, **stop** stop editor 1-668 to 1-668
 ex command, **substitute** replace string 1-668 to 1-668
 ex command, **suspend** stop edit 1-668 to 1-668
 ex command, **t** copy text 1-668 to 1-668
 ex command, **tag** tag name 1-668 to 1-668
 ex command, **unabbreviate** delete abbreviation 1-669 to 1-669
 ex command, **undo** undo last change 1-669 to 1-669
 ex command, **unmap** delete macro 1-669 to 1-669
 ex command, **v** global on non-matching lines 1-669 to 1-669
 ex command, **version** editor version 1-669 to 1-669
 ex command, **visual** enter vi mode 1-669 to 1-669
 ex command, **wn** write and edit next 1-669 to 1-669
 ex command, **wq** write and quit 1-670 to 1-670
 ex command, **write** write file 1-669 to 1-669
 ex command, **xit** write and quit 1-670 to 1-670
 ex command, **yank** yank text to buffer 1-670 to 1-670
 ex command, **z** display lines 1-670 to 1-670
 ex command, **CTRL-D** display lines 1-671 to 1-671
 ex command, **~** repeat last substitute 1-671 to 1-671
 ex edit new file ex command 1-664 to 1-664
 ex mode 1-646 to 1-663
 ex variable, **prompt** prompt for ex input 1-674 to 1-674
 ex variable, **shell** choose command interpreter 1-668 to 1-675

ex, alternate files in 1-672 to 1-672
 examples, awk 1-32 to 1-32
 exec command 1-210 to 1-213
 execute buffer ex command, @ 1-671 to 1-671
 executing strings, dc 1-161 to 1-161
 execution environment, shell 1-539 to 1-539
 EXINIT environment variable 1-647 to 1-678
 exit command 1-214 to 1-214
 exit status 1-118 to 1-617
 expand command 1-215 to 1-216
 expansion, keyword 1-109 to 1-113
 experimental state 1-458 to 1-98
 exponentiation, dc 1-158 to 1-158
 export command 1-217 to 1-218, 3-3 to 3-3
 exporting aliases 1-9 to 1-9
 exporting environment variables 1-217 to 1-217
 expr command 1-219 to 1-222, 3-32 to 3-37
 expr operators 1-219 to 1-219
 expression evaluation 1-219 to 1-219
 expression, test conditional 1-598 to 1-598
 exrc ignore .exrc file vi variable 1-673 to 1-673
 exrecover command 1-666 to 1-666
 extraction, archive 1-416 to 1-416

F

F search backwards for character vi command 1-652 to 1-652
f search forward for character vi command 1-652 to 1-652
 failure 1-223 to 1-223
 false command 1-223 to 1-223
 fc command 1-10 to 1-226, 3-38 to 3-45
 FCEDIT environment variable 1-225 to 1-542
 FCOPY command 1-243 to 1-243
 fg command 1-274 to 1-658
 fgrep command 1-259 to 1-535
 Fibonacci sequence 1-164 to 1-541
 field in current record, awk NF 1-24 to 1-24
 field separator, internal 1-543 to 1-543
 fields 1-147 to 1-147
 FIFO 1-374 to 1-376
 file attributes 1-300 to 1-88
 file being read, awk FILENAME 1-24 to 1-24
 file command 1-227 to 1-230, 2-4 to 2-5
 file descriptor 1-536 to 1-599
 file format 1-458 to 1-470
 file format, cpio 1-137 to 1-595, 2-10 to 2-3
 file format, ctags 1-687 to 1-687
 file format, magic 1-228 to 1-230, 2-4 to 2-5
 file format, pax 1-431 to 1-431, 2-10 to 2-6
 file format, tags 1-143 to 1-146, 2-7 to 2-7
 file format, tar 1-137 to 1-595, 2-3 to 2-10
 file information vi command, CTRL-G 1-657 to 1-657
 file mode 1-110 to 1-98
 file mode, change 1-88 to 1-88
 file name generation 1-193 to 1-538
 file name suffix 1-44 to 1-456
 file name, .exrc 1-647 to 1-678
 file name, .mailrc 1-319 to 1-319

file name, .profile A-18 to A-18, A-2 to A-2, A-2 to A-2, A-2 to A-2, 1-517 to 1-545
file name, .sh_history A-2 to A-2, 1-267 to 1-545
file name, a.out 1-71 to 1-73
file name, calendar 1-76 to 1-77
file name, CCOMXL.PUB.SYS 1-72 to 1-72
file name, dead.letter 1-322 to 1-326
file name, ed.hup 1-202 to 1-202
file name, l.output 1-288 to 1-288
file name, lex.yy.c 1-287 to 1-288
file name, LINKEDIT.PUB.SYS 1-72 to 1-72
file name, magic 2-4 to 2-4
file name, makefile 1-336 to 1-358
file name, Makefile 1-336 to 1-358
file name, man.dbz 1-363 to 1-365
file name, mbox 1-306 to 1-306
file name, rsh 1-519 to 1-519
file name, sh 1-519 to 1-519
file name, startup.mk 1-342 to 1-359
file name, states.out 1-711 to 1-711
file name, tags 1-143 to 1-668, 2-7 to 2-7
file name, whatis 1-364 to 1-365
file name, y.output 1-711 to 1-711
file name, y.tab.c 1-710 to 1-711
file name, y.tab.h 1-710 to 1-711
file obtain file information **ex** command 1-664 to 1-664
file owner 1-109 to 1-95
file **pax** archive, component 1-416 to 1-416
file permissions 1-233 to 1-386
file transmission 1-641 to 1-644
file type 1-227 to 1-234
file, RCS 1-109 to 1-99
file, reject 1-409 to 1-409
file, shared 1-457 to 1-458
file, special 1-376 to 1-376
file, working 1-109 to 1-99
file-processing 1-17 to 1-17
FILENAME file being read, **awk** 1-24 to 1-24
files in **ex**, alternate 1-672 to 1-672
files, archive 1-133 to 1-416
files, break up 1-567 to 1-567
files, byte stream 1-126 to 1-690
files, compare sorted 1-119 to 1-119
files, compare three 1-181 to 1-181
files, compare two 1-174 to 1-185
files, compressing 1-402 to 1-637
files, display 1-433 to 1-81
files, dumping 1-397 to 1-397
files, encoding 1-402 to 1-637
files, first lines of 1-263 to 1-263
files, list 1-299 to 1-299
files, merge 1-434 to 1-559
files, moving 1-386 to 1-386
files, packed 1-432 to 1-432
files, paginate 1-433 to 1-433
files, remove 1-494 to 1-494
files, search for 1-231 to 1-231
files, shell 1-44 to 1-44
files, sorted 1-119 to 1-119
files, sorting 1-559 to 1-559
files, splitting 1-138 to 1-567
files, system 1-598 to 1-598
files, temporary 1-113 to 1-99
files, text 1-174 to 1-98
filter **vi** command, ! 1-655 to 1-655
find command 1-133 to 1-237
first lines of files 1-263 to 1-263
fixed to variable record conversion 1-170 to 1-170
flag, quiet 1-111 to 1-95
flipr environment variable 1-313 to 1-324
fmt command 1-238 to 1-240
fnmatch function 3-32 to 3-37
FNR number of records read from file, **awk** 1-24 to 1-24
FNR variable 1-25 to 1-25
fold command 1-241 to 1-242
folder environment variable 1-309 to 1-330
folder, mail 1-322 to 1-322
folding lines 1-241 to 1-241
font 1-42 to 1-42
font, italic 1-42 to 1-42
fontfile 1-42 to 1-42
for shell command 1-127 to 1-68
format, date 1-110 to 1-110
format, **ex** command 1-659 to 1-659
formatting, text 1-238 to 1-238
FORTRAN 1-15 to 1-15
forwarding mail 1-306 to 1-315
frombyte command A-3 to A-3, 1-243 to 1-244
FS variable 1-18 to 1-27
full word 1-651 to 1-651
function shell command 1-524 to 1-524
function, **fnmatch** 3-32 to 3-37
function, **getlogin** 1-298 to 1-298
function, **iconv** 3-5 to 3-5
function, **iconv_open** 3-5 to 3-5
function, **input** 3-6 to 3-7
function, **length** 1-59 to 1-59
function, **main** 1-143 to 1-143
function, **output** 3-6 to 3-6
function, **printf** 1-150 to 1-150
function, **regcomp** 3-31 to 3-37
function, **regerror** 1-261 to 1-261, 3-31 to 3-31
function, **regex** 3-31 to 3-31
function, **regfree** 3-31 to 3-31
function, **scale** 1-48 to 1-59
function, **shell** 1-486 to 1-621
function, **sprintf** 1-39 to 1-39
function, **stat** 2-9 to 2-9
function, **symlink** 1-427 to 1-427
function, **tcgetattr** 1-580 to 1-580
function, **tcsetattr** 1-580 to 1-580
function, **unput** 3-7 to 3-7
function, **vfprintf** 3-7 to 3-7
function, **wordexp** 1-550 to 1-552
function, **yycomment** 3-7 to 3-7
function, **yyerror** 3-7 to 3-7
function, **yygetc** 3-7 to 3-7
function, **yyless** 3-8 to 3-8

function, `yylex` 3-6 to 3-8
function, `yymapch` 3-7 to 3-7
function, `yymore` 3-8 to 3-8
function, `yywrap` 3-7 to 3-7
functions 1-45 to 1-45
functions command 1-10 to 1-246
functions, `awk` 1-28 to 1-28

G

`g` global find `ex` command 1-664 to 1-664
G move to absolute line `vi` command 1-650 to 1-650
generation, file name 1-193 to 1-538
`getconf` command 1-154 to 1-252
`getline`, `awk` 1-25 to 1-25
`getlogin` function 1-298 to 1-298
`getopt` command 1-253 to 1-255
`getopts` command 1-255 to 1-258
`glob` command 3-32 to 3-32
global `ed` command 1-198 to 1-198
global find `ex` command, `g` 1-664 to 1-664
global on non-matching lines `ex` command, `v` 1-669 to 1-669
`gmacs` 1-516 to 1-542
grammar 1-710 to 1-710
`grep` command 1-206 to 1-261, 3-32 to 3-37
`grep` matching strings 1-259 to 1-259
`grep` patterns 1-259 to 1-259
`grep` searching for strings 1-259 to 1-259
`gres` command 1-261 to 1-261
group directory 1-496 to 1-93
group ID 1-86 to 1-86
group, home 1-77 to 1-77
groups 1-232 to 1-269
`GROUPSHELL` macro 1-356 to 1-356
gutter 1-69 to 1-69

H

`h` move left `vi` command 1-650 to 1-650
H move to top of screen `vi` command 1-652 to 1-652
`hash` command 1-10 to 1-262
`head` command 1-263 to 1-264
head revision 1-488 to 1-489
header 1-113 to 1-490
header environment variable 1-323 to 1-323
Header keyword 1-113 to 1-271
headers, mail 1-304 to 1-325
`help` command 1-265 to 1-266
`HELP` environment variable 1-265 to 1-265
`HELPCMD` environment variable 1-265 to 1-265
`helpfile` path name 1-265 to 1-265
`helpindex` path name 1-265 to 1-265
here document 1-537 to 1-537
highest branch 1-488 to 1-488
`HISTFILE` environment variable 1-224 to 1-548
`history` command 1-10 to 1-268
history file 1-224 to 1-543
history, command 1-224 to 1-453

`HISTSIZ` environment variable 1-224 to 1-601
hold buffer 1-506 to 1-506
`hold` environment variable 1-307 to 1-323
home directory 1-320 to 1-83
`HOME` environment variable A-2 to A-2, A-2 to A-2, 1-201 to 1-84, 3-3 to 3-3
home group 1-77 to 1-77
`HPLDEVNO` variable 1-619 to 1-619
`HPQUIET` variable 1-371 to 1-371
Huffman encoding 1-402 to 1-637

I

I insert at start of line `vi` command 1-656 to 1-656
`i` insert text `vi` command 1-656 to 1-656
`ibase` variable 1-48 to 1-66
`iconv` 1-419 to 1-431, 3-5 to 3-5
`iconv` function 3-5 to 3-5
`iconv_open` function 3-5 to 3-5
`id` command 1-269 to 1-270
ID keyword 1-113 to 1-271
ID, group 1-86 to 1-86
`id`, parent process 1-544 to 1-544
`id`, process 1-544 to 1-544
ID, user 1-92 to 1-92
`ident` command 1-104 to 1-272, 3-26 to 3-26
identifier 1-458 to 1-458, 3-25 to 3-25
`if` shell command 1-521 to 1-615
`IFS` environment variable 1-478 to 1-543
ignore `.exrc` file `vi` variable, `exrc` 1-673 to 1-673
ignore case in pattern matching `vi` variable, `ignorecase` 1-673 to 1-673
ignore configuration options 1-467 to 1-467
ignore environment variable 1-305 to 1-323
ignore white space 1-174 to 1-174
`ignorecase` ignore case in pattern matching `vi` variable 1-673 to 1-673
`ignoreeof` environment variable 1-322 to 1-323
implementation considerations A-1 to A-19
`in` shell command 1-524 to 1-524
include file, `signal.h` A-14 to A-14
increment autoindent `vi` command, `CTRL-T` 1-658 to 1-658
`indent` environment variable 1-323 to 1-323
indentation 1-239 to 1-239
indentation distance `vi` variable, `shiftwidth` 1-655 to 1-676
`indentprefix` environment variable 1-317 to 1-323
initial revision 1-95 to 1-97
initialize RCS file 1-457 to 1-98
`input` function 3-6 to 3-7
`input` mode 1-305 to 1-332
`input`, `nroff` 1-227 to 1-227
insert at start of line `vi` command, **I** 1-656 to 1-656
insert character literally `vi` command, `CTRL-Q` 1-658 to 1-658
insert character literally `vi` command, `CTRL-V` 1-658 to 1-678
`insert` command 1-211 to 1-211
`insert` `ed` command 1-198 to 1-198

insert insert text *ex* command 1-665 to 1-665
insert mode 1-646 to 1-654
insert mode *vi* 1-659 to 1-659
insert text *ex* command, **insert** 1-665 to 1-665
insert text *vi* command, **i** 1-656 to 1-656
insertion commands, *vi* text 1-648 to 1-648
integer command 1-10 to 1-273
interactive shell 1-520 to 1-520
intercept interrupts 1-615 to 1-615
internal field separator 1-543 to 1-543
Internet network 1-305 to 1-305
interrupts, intercept 1-615 to 1-615
interrupts, trap 1-615 to 1-615
intro command 1-1 to 1-1
invoke *ex* command from *vi* command, **:** 1-657 to 1-657
invoke macro *vi* command, **@** 1-657 to 1-657
invoke subshell *ex* command, **shell** 1-668 to 1-668
invoke system command, **~!** 3-9 to 3-9
invoke text editor on message, **~e** 3-9 to 3-9
invoke visual editor on message, **~v** 3-9 to 3-9
italic font 1-42 to 1-42

J

J join lines *vi* command 1-656 to 1-656
j move down one line *vi* command 1-650 to 1-650
job control 1-275 to 1-518
jobs command 1-274 to 1-275
join command 1-276 to 1-279
join databases 1-276 to 1-276
join *ed* command 1-198 to 1-198
join join lines *ex* command 1-665 to 1-665
join lines *ex* command, **join** 1-665 to 1-665
join lines *vi* command, **J** 1-656 to 1-656
join RCS revisions 1-110 to 1-111
joinlist 1-110 to 1-110
justified margins 1-238 to 1-238

K

k mark text *ex* command 1-665 to 1-665
k move up one line *vi* command 1-650 to 1-650
keep environment variable 1-323 to 1-323
keepsave environment variable 1-306 to 1-323
key, sorting 1-561 to 1-561
keyword expansion 1-109 to 1-113
keyword value 1-112 to 1-97
keyword, Author 1-113 to 1-271
keyword, Date 1-113 to 1-271
keyword, Header 1-113 to 1-271
keyword, Id 1-113 to 1-271
keyword, Locker 1-113 to 1-271
keyword, Log 1-113 to 1-456
keyword, Name 1-113 to 1-113
keyword, RCSfile 1-113 to 1-271
keyword, Revision 1-113 to 1-271
keyword, Source 1-113 to 1-271
keyword, State 1-113 to 1-271
kill command A-16 to A-16, A-16 to A-16, 1-274 to 1-282

L

L move right *vi* command 1-650 to 1-650
L move to bottom of screen *vi* command 1-652 to 1-652
l.output file name 1-288 to 1-288
LALR(1) 1-710 to 1-710
last lines of file 1-588 to 1-588
lc command 1-121 to 1-284
LC_COLLATE environment variable 1-249 to 1-612, 3-30 to 3-30
LC_CTYPE environment variable 1-612 to 1-613, 3-30 to 3-30
LC_MESSAGES environment variable 1-233 to 1-494
LC_TIME environment variable 1-152 to 1-152
leave insert mode *vi* command, **ESC** 1-659 to 1-659
Lempel-Ziv compression 1-123 to 1-629
length function 1-59 to 1-59
length of tags *vi* variable, taglength 1-669 to 1-676
let command 1-10 to 1-286
let macros contain macros *vi* variable, remap 1-665 to 1-675
lex 1-287 to 1-291, 3-6 to 3-8
lex command 1-227 to 1-291, 3-6 to 3-8
lex DFA table 1-287 to 1-287
lex move table 1-287 to 1-287
lex NFA table 1-287 to 1-287
lex statement, **ECHO** 3-8 to 3-8
lex statement, **REJECT** 3-8 to 3-8
lex.yy.c file name 1-287 to 1-288
lexical analyzer 1-287 to 1-287
libraries 1-11 to 1-11
libraries, archive 1-11 to 1-11
line command 1-292 to 1-292
LINE environment variable 1-384 to 1-384
line width 1-238 to 1-239
line, absolute 1-650 to 1-650
LINENO environment variable 1-543 to 1-554
LINES environment variable 1-382 to 1-678, 3-3 to 3-3
lines, center 1-238 to 1-238
lines, concatenate 1-405 to 1-405
lines, copy 1-292 to 1-292
lines, count 1-694 to 1-694
lines, folding 1-241 to 1-241
lines, numbering 1-393 to 1-393
lines, read 1-478 to 1-478
lines, reverse 1-487 to 1-487
lines, unique 1-634 to 1-634
LINE_MAX variable 1-261 to 1-261
link 1-293 to 1-293
link editor, MPEIX 1-13 to 1-75
linkedit command 1-74 to 1-74
LINKEDIT.PUB.SYS file name 1-72 to 1-72
list directory 1-283 to 1-299
list display lines in verbose form *ex* command 1-665 to 1-665
list display lines in verbose form *vi* variable 1-674 to 1-674
list files 1-299 to 1-299
list lines *ed* command 1-199 to 1-199
list, process 1-274 to 1-274

LISTER environment variable 1-323 to 1-323
LISTFILE command A-17 to A-17, A-17 to A-17
LISTUSER command 1-77 to 1-77
ln command 1-293 to 1-297
local time 1-110 to 1-110
local variables 1-621 to 1-621
locale command 1-297 to 1-297
lock 1-109 to 1-97
lock revision 1-457 to 1-490
lock, strict 1-457 to 1-98
locker 1-271 to 1-489
Locker keyword 1-113 to 1-271
locking, non-strict 1-458 to 1-458
locking, strict 1-113 to 1-95
locks, breaking 1-458 to 1-95
Log keyword 1-113 to 1-456
log message 1-113 to 1-97
log, RCS 1-456 to 1-456
logging in 1-298 to 1-700
login command 1-320 to 1-320, 3-3 to 3-3
login name 1-112 to 1-98
login names 1-309 to 1-324
login shell 1-224 to 1-521
logname command 1-270 to 1-298
LOGNAME environment variable 1-298 to 1-320, 3-3 to 3-3
long type 2-4 to 2-4
loop 1-127 to 1-68
lowercase conversion 1-169 to 1-169
ls command A-13 to A-13, 1-106 to 1-303

M

m mark current position **vi** command 1-650 to 1-650
M move to middle of screen **vi** command 1-652 to 1-652
machine configuration 1-627 to 1-627
macro 1-530 to 1-675
macro, GROUPSHELL 1-356 to 1-356
macro, MAKEFLAGS 1-350 to 1-350
macro, SHELL 1-353 to 1-359
macro, yygetc 3-7 to 3-7
magic file format 1-228 to 1-230, 2-4 to 2-5
magic file name 2-4 to 2-4
magic use pattern matching metacharacters **vi** variable 1-661 to 1-674
mail 1-543 to 1-641
mail command 1-325 to 1-465
MAIL environment variable 1-543 to 1-543
mail folder 1-322 to 1-322
mail headers 1-304 to 1-325
mail server 1-322 to 1-322
mail, electronic 1-304 to 1-304
mail, forwarding 1-306 to 1-315
mail, send 1-458 to 1-458
MAILCHECK environment variable 1-543 to 1-601
MAILDIR environment variable 1-320 to 1-326
MAILER environment variable 1-458 to 1-77, 3-3 to 3-3
MAILPATH environment variable 1-527 to 1-543
MAILRC environment variable 1-319 to 1-332
mailx command 1-240 to 1-333, 3-3 to 3-55

main function 1-143 to 1-143
make command 1-14 to 1-361
Makefile file name 1-336 to 1-358
makefile file name 1-336 to 1-358
MAKEFLAGS environment variable 1-350 to 1-355
MAKEFLAGS macro 1-350 to 1-350
MAKESTARTUP environment variable 1-335 to 1-353
man command 1-266 to 1-366
man.dbz file name 1-363 to 1-365
manipulation commands, vi 1-648 to 1-648
manipulation vi, object 1-653 to 1-653
manipulator abbreviations, object 1-655 to 1-655
MANPAGER environment variable 1-363 to 1-365
MANPATH environment variable 1-362 to 1-365
manual pages 1-362 to 1-362
map define macro **vi** command 1-665 to 1-665
margins, justified 1-238 to 1-238
mark current position vi command, **m** 1-650 to 1-650
mark ed command 1-199 to 1-199
mark mark text **ex** command 1-665 to 1-665
mark name vi command 1-650 to 1-650
mark text ex command, **k** 1-665 to 1-665
mark text ex command, **mark** 1-665 to 1-665
matching strings, grep 1-259 to 1-259
maxbuffers buffer size **vi** variable 1-674 to 1-674
MBOX environment variable 1-323 to 1-326
mbox file name 1-306 to 1-306
merge command 1-367 to 1-369
merge files 1-434 to 1-559
merge RCS revisions 1-474 to 1-474
msg command 1-370 to 1-371
msg turn on talk and write permissions **vi** variable 1-674 to 1-674
msg variable 1-674 to 1-674
metacharacter 1-661 to 1-674, 3-32 to 3-32
metoo environment variable 1-324 to 1-324
minimal differences 1-175 to 1-175
mkdir command A-12 to A-12, 1-372 to 1-373
mkfifo command 1-374 to 1-375
mknod command A-19 to A-19, 1-375 to 1-377
mode vi, insert 1-659 to 1-659
mode, command 1-307 to 1-646
mode, ex 1-646 to 1-663
mode, file 1-110 to 1-98
mode, input 1-305 to 1-332
mode, insert 1-646 to 1-654
mode, quiet 1-195 to 1-97
mode, vi 1-646 to 1-657
modification time 1-128 to 1-299
modification time, change 1-605 to 1-605
modulo, dc 1-158 to 1-158
month sort 1-559 to 1-559
more command 1-143 to 1-385, 2-7 to 2-7
MORE environment variable 1-382 to 1-382
move back to start of paragraph vi command, **{** 1-652 to 1-652
move back to start of section vi command, **[[** 1-653 to 1-653
move down one line vi command, **j** 1-650 to 1-650

move down one line vi command, **CTRL-J** 1-650 to 1-650
 move down one line vi command, **CTRL-N** 1-650 to 1-650
 move ed command 1-199 to 1-199
 move left vi command, backspace 1-650 to 1-650
 move left vi command, **h** 1-650 to 1-650
 move left vi command, **CTRL-H** 1-650 to 1-650
move move text ex command 1-665 to 1-665
 move right vi command, **l** 1-650 to 1-650
 move right vi command, space 1-650 to 1-650
 move table entries, number of 1-287 to 1-287
 move table, lex 1-287 to 1-287
 move text ex command, **move** 1-665 to 1-665
 move to absolute line vi command, **G** 1-650 to 1-650
 move to balancing bracket vi command, **%** 1-653 to 1-653
 move to bottom of screen vi command, **L** 1-652 to 1-652
 move to column vi command, **|** 1-651 to 1-651
 move to end of full word vi command, **E** 1-651 to 1-651
 move to end of line vi command, **\$** 1-651 to 1-651
 move to end of word vi command, **e** 1-652 to 1-652
 move to first non-blank vi command, **^** 1-651 to 1-651
 move to mark vi command, **`** 1-651 to 1-651
 move to middle of screen vi command, **M** 1-652 to 1-652
 move to next full word vi command, **W** 1-652 to 1-652
 move to next paragraph vi command, **}** 1-652 to 1-652
 move to next section vi command, **]]** 1-653 to 1-653
 move to next sentence vi command, **)** 1-652 to 1-652
 move to next word vi command, **w** 1-652 to 1-652
 move to previous full word vi command, **B** 1-651 to 1-651
 move to previous sentence vi command, **(** 1-652 to 1-652
 move to previous word vi command, **b** 1-651 to 1-651
 move to start of line vi command, **O** 1-650 to 1-650
 move to start of marked line vi command, **^** 1-651 to 1-651
 move to start of next line vi command, **+** 1-650 to 1-650
 move to start of next line vi command, **CTRL-M** 1-650 to 1-650
 move to start of previous line vi command, **-** 1-651 to 1-651
 move to top of screen vi command, **H** 1-652 to 1-652
 move up one line vi command, **k** 1-650 to 1-650
 move up one line vi command, **CTRL-P** 1-650 to 1-650
 movement commands, vi 1-648 to 1-648
 movement, vi absolute 1-648 to 1-650
 movement, vi context-dependent 1-648 to 1-651
 moving files 1-386 to 1-386
 MPE/IX CI 1-13 to 1-77
 MPE/IX compiler 1-75 to 1-75
 MPE/IX link editor 1-13 to 1-75
 multi-column 1-69 to 1-69
 multi-line comments 1-456 to 1-456
 multiple commands in ex, | 1-659 to 1-659
 multiple volume support 1-135 to 1-421
 multiplication, dc 1-158 to 1-158
 mv command 1-132 to 1-390

N

n repeat string search vi command 1-652 to 1-652
N reverse string search vi command 1-652 to 1-652
 Name keyword 1-113 to 1-113

name tags file vi variable, tags 1-668 to 1-676
 name, login 1-112 to 1-98
 name, symbolic revision 1-111 to 1-97
 name, user 1-269 to 1-269
 named buffer 1-654 to 1-654
 names, command 1-696 to 1-698
 names, login 1-309 to 1-324
 network, Internet 1-305 to 1-305
 new process 1-212 to 1-212
 NEWACCT command A-11 to A-11, A-11 to A-11
 newline 1-647 to 1-647
 NEWUSER command A-6 to A-6
next edit next file ex command 1-666 to 1-666
 NF field in current record, awk 1-24 to 1-24
 NF variable 1-25 to 1-25
 NFA table entries, number of 1-287 to 1-287
 NFA table, lex 1-287 to 1-287
 nice command 1-391 to 1-392
 nl command 1-393 to 1-396
 noallnet environment variable 1-321 to 1-321
 noappend environment variable 1-321 to 1-321
 noask environment variable 1-321 to 1-321
 noaskbcc environment variable 1-321 to 1-321
 noaskcc environment variable 1-321 to 1-321
 noasksub environment variable 1-321 to 1-321
 noautoprint environment variable 1-320 to 1-321
 nobang environment variable 1-321 to 1-321
 nocrt environment variable 1-322 to 1-322
 nodot environment variable 1-322 to 1-322
 noflipr environment variable 1-322 to 1-322
 nohold environment variable 1-323 to 1-323
 nohup command 1-392 to 1-392
 noignore environment variable 1-323 to 1-323
 noignoreeof environment variable 1-323 to 1-323
 nokeep environment variable 1-323 to 1-323
 nokeepsave environment variable 1-323 to 1-323
 nometoo environment variable 1-324 to 1-324
 non-strict locking 1-458 to 1-458
 nooutfolder environment variable 1-324 to 1-324
 nopage environment variable 1-324 to 1-324
 noquiet environment variable 1-324 to 1-324
 norecord environment variable 1-324 to 1-324
 nosave environment variable 1-325 to 1-325
 nosendwait environment variable 1-325 to 1-325
 noshowto environment variable 1-325 to 1-325
 noSign environment variable 1-325 to 1-325
 nosign environment variable 1-325 to 1-325
 NR number of records read, awk 1-24 to 1-24
 NR variable 1-25 to 1-25
 nroff input 1-227 to 1-227
 nroff program 1-175 to 1-675
number display lines with line numbers ex command 1-666 to 1-666
 number display lines with line numbers vi variable 1-674 to 1-674
 number lines ed command 1-199 to 1-199
 number of DFA table entries 1-287 to 1-287
 number of move table entries 1-287 to 1-287
 number of NFA table entries 1-287 to 1-287

number of records read from file, `awk` `FNR` 1-24 to 1-24
number of records read, `awk` `NR` 1-24 to 1-24
number, revision 1-111 to 1-95
numbered buffer 1-654 to 1-654
numbering lines 1-393 to 1-393
numbering, page 1-394 to 1-394
numbers, command 1-225 to 1-225

O

o open new line after current `vi` command 1-656 to 1-656
O open new line before current `vi` command 1-656 to 1-656
obase variable 1-48 to 1-49
object manipulation `vi` 1-653 to 1-653
object manipulator abbreviations 1-655 to 1-655
obtain file information `ex` command, **file** 1-664 to 1-664
`od` command 1-397 to 1-401
`OFMT` variable 1-18 to 1-31
`OFS` variable 1-31 to 1-31
`OLDPWD` environment variable 1-529 to 1-83
`onehop` environment variable 1-324 to 1-333
online documentation 1-362 to 1-362
open enter open mode `ex` command 1-666 to 1-666
open new line after current `vi` command, **o** 1-656 to 1-656
open new line before current `vi` command, **O** 1-656 to 1-656
operators, `awk` 1-21 to 1-21
operators, `expr` 1-219 to 1-219
`OPTARG` variable 1-256 to 1-257
`OPTIND` environment variable 1-546 to 1-601
`OPTIND` variable 1-256 to 1-257
options, command line 1-253 to 1-253
`ORS` variable 1-31 to 1-31
`out` environment variable 1-537 to 1-537
outdated revision 1-458 to 1-458
`outfolder` environment variable 1-324 to 1-324
output function 3-6 to 3-6
output stream, duplicate 1-596 to 1-596
output stream, `split` 1-596 to 1-596
output, double-spaced 1-433 to 1-433
overlap 1-109 to 1-475
owner of RCS file 1-457 to 1-457
owner, file 1-109 to 1-95

P

p paste after `vi` command 1-657 to 1-657
P paste before `vi` command 1-656 to 1-656
`pack` command 1-126 to 1-404
packed files 1-432 to 1-432
`page` environment variable 1-312 to 1-324
page numbering 1-394 to 1-394
page width 1-69 to 1-69
`PAGER` environment variable 1-313 to 1-364
pages 1-393 to 1-393
pages, manual 1-362 to 1-362
paginate files 1-433 to 1-433
paragraph 1-652 to 1-652

paragraphs paragraph macro names `vi` variable 1-652 to 1-674
parameter substitution 1-529 to 1-529
parameters, positional 1-529 to 1-556
parameters, shift positional 1-556 to 1-556
parent process id 1-544 to 1-544
parsing command line options 1-253 to 1-253
`passwd` 2-10 to 2-10
paste after `vi` command, **p** 1-657 to 1-657
paste before `vi` command, **P** 1-656 to 1-656
`paste` command 1-149 to 1-407
paste lines `ex` command, **put** 1-666 to 1-666
`patch` command 1-180 to 1-413
`PATH` environment variable 1-121 to 1-9, 3-3 to 3-3
path name, / A-10 to A-10
path name, /bin/sh 1-667 to 1-667
path name, /dev/mt/0m 1-591 to 1-591
path name, /etc 1-265 to 1-265
path name, /etc/helpfile 1-266 to 1-266
path name, /etc/helpindex 1-266 to 1-266
path name, /etc/init 1-700 to 1-700
path name, /etc/italic.fnt 1-42 to 1-42
path name, /etc/magic 1-227 to 1-227, 2-4 to 2-4
path name, /etc/mailx.rc 1-305 to 1-333
path name, /etc/passwd 1-529 to 1-529
path name, /etc/profile A-18 to A-18, 1-520 to 1-555, 3-4 to 3-4
path name, /etc/small.fnt 1-42 to 1-42
path name, /etc/startup.mk 1-351 to 1-353
path name, /etc/suid_profile 1-517 to 1-545
path name, /etc/utmp 1-700 to 1-701
path name, /etc/wtmp 1-700 to 1-701
path name, /etc/yylex.c 1-287 to 1-288
path name, /etc/yyparse.c 1-710 to 1-711
path name, /tmp 1-179 to 1-572, 3-4 to 3-4
path name, /tmp/e(** 1-202 to 1-202
path name, /tmp/sh(** 1-545 to 1-545
path name, /tmp/stm(** 1-563 to 1-563
path name, /usr/lib/lib.b 1-63 to 1-63
path name, /usr/lib/libl.a 1-288 to 1-288, 3-8 to 3-8
path name, /usr/mail 1-320 to 1-326
path name, /usr/man 1-362 to 1-364
path name, /usr/man/man1/(** .1 1-266 to 1-266
path name, calendar 1-76 to 1-76
path name, conf.c 1-376 to 1-376
path name, conflicting 1-293 to 1-293
path name, helpfile 1-265 to 1-265
path name, helpindex 1-265 to 1-265
path name, pk\$(** 1-402 to 1-637
path name, RCS 1-104 to 1-499
path prefix 1-457 to 1-457
path, search 1-526 to 1-698
`pathchk` command 1-187 to 1-415
pattern buffer 1-506 to 1-506
patterns 1-133 to 1-523
patterns, `awk` 1-29 to 1-29
patterns, `grep` 1-259 to 1-259
pax archive, component directory 1-416 to 1-416

pax archive, component file 1-416 to 1-416
 pax command A-19 to A-19, 1-126 to 1-431, 2-3 to 2-6, 3-5 to 3-5
 pax file format 1-431 to 1-431, 2-10 to 2-6
 pcat command 1-363 to 1-432
 permission, read 1-113 to 1-99
 permissions, file 1-233 to 1-386
 phone lines 1-644 to 1-644
 PID 1-544 to 1-544
 PID environment variable 1-544 to 1-544
 pipe 1-374 to 1-667
 pipe ex command 1-669 to 1-669
 pipeline 1-111 to 1-522
 pk\$ (** path name 1-402 to 1-637
 position pointer, current 1-648 to 1-648
 positional parameters 1-529 to 1-556
 positional parameters, shift 1-556 to 1-556
 POSIX.1 1-247 to 1-247
 PPID 1-544 to 1-544
 pr command 1-216 to 1-436
 precedence 1-49 to 1-49
 precedes error message with alert vi variable, errorbells 1-673 to 1-673
 prefix, path 1-457 to 1-457
 prefix, program segment 1-544 to 1-544
 preprocessor, C 1-175 to 1-175
 preserve command 1-667 to 1-667
preserve saves buffer for recovery ex command 1-666 to 1-666
 previous context 1-651 to 1-651
 previous directory 1-544 to 1-544
 previous working directory 1-83 to 1-83
 print arguments 1-437 to 1-437
 print command 1-226 to 1-438
print display lines ex command 1-666 to 1-666
 print lines ed command 1-199 to 1-199
 printenv command 1-208 to 1-208
 printf command 1-194 to 1-441, 2-4 to 2-4
 printf function 1-150 to 1-150
 priority 1-391 to 1-391
 privileged shell 1-545 to 1-545
 process 1-274 to 1-274
 process id 1-544 to 1-544
 process id, parent 1-544 to 1-544
 process list 1-274 to 1-274
 process, new 1-212 to 1-212
 process, terminate 1-280 to 1-280
 program segment prefix 1-544 to 1-544
 program, nroff 1-175 to 1-675
 program, troff 1-175 to 1-675
 programming, C 1-143 to 1-143
 programs, assembler 1-227 to 1-227
 prompt ed command 1-199 to 1-199
 prompt environment variable 1-324 to 1-324
 prompt for ex input ex variable, prompt 1-674 to 1-674
 prompt prompt for ex input ex variable 1-674 to 1-674
 prompt string, ed 1-195 to 1-195
 prompt string, sh 1-544 to 1-544
 ps command 1-281 to 1-450

PS1 environment variable 1-527 to 1-544
 PS2 environment variable 1-479 to 1-544
 PS3 environment variable 1-525 to 1-554
 PS4 environment variable 1-527 to 1-544
 PSP 1-544 to 1-544
put paste lines ex command 1-666 to 1-666
 pwd command 1-10 to 1-452
 PWD environment variable 1-516 to 1-83

Q

Q enter ex mode vi command 1-657 to 1-657
 quiet environment variable 1-324 to 1-324
 quiet flag 1-111 to 1-95
 quiet mode 1-195 to 1-97
 quiet mode ed command 1-198 to 1-198
 quiet mode vi variable, quiet 1-675 to 1-675
 quiet quiet mode vi variable 1-675 to 1-675
 quit ed command 1-199 to 1-199
 quit editing session ex command, **quit** 1-666 to 1-666
quit quit editing session ex command 1-666 to 1-666
 quoting special characters 1-528 to 1-528

R

r command 1-10 to 1-455
r replace character vi command 1-656 to 1-656
R type over characters vi command 1-656 to 1-656
 RANDOM environment variable 1-544 to 1-544
 range, date 1-488 to 1-488
 range, revision 1-457 to 1-489
 RCS 1-104 to 1-505
 rcs command 1-102 to 1-465, 3-10 to 3-26
 RCS command 1-460 to 1-460
 RCS configuration file 1-458 to 1-458
 RCS file 1-109 to 1-99
 RCS file, create 1-457 to 1-95
 RCS file, initialize 1-457 to 1-98
 RCS file, owner of 1-457 to 1-457
 RCS log 1-456 to 1-456
 RCS path name 1-104 to 1-499
 RCS revision, compare 1-470 to 1-470
 RCS revisions, join 1-110 to 1-111
 RCS revisions, merge 1-474 to 1-474
 RCS, undo change 1-475 to 1-475
 rcs clean command 1-104 to 1-469, 3-26 to 3-26
 rcs diff command 1-104 to 1-473, 3-26 to 3-26
 rcs edit 1-103 to 1-98, 3-15 to 3-10
 rcs error 1-116 to 1-6, 3-10 to 3-23
 rcs file 1-104 to 1-505, 3-16 to 3-26
 RCS file keyword 1-113 to 1-271
 rcsid 1-271 to 1-271
 rcs merge command 1-104 to 1-477, 3-26 to 3-26
 RCSPATH environment variable 1-103 to 1-501, 3-12 to 3-15
 read command 1-226 to 1-480
 read file ed command 1-199 to 1-199
 read file ex command, **read** 1-667 to 1-667
 read file into message, ~r 3-9 to 3-9

read lines 1-478 to 1-478
 read permission 1-113 to 1-99
read read file *ex* command 1-667 to 1-667
 read-only 1-481 to 1-481
 read-only file 1-283 to 1-283
 readonly command 1-481 to 1-482
 readonly do not write on file *vi* variable 1-647 to 1-675
 record environment variable 1-304 to 1-324
 recover file *ex* command, **recover** 1-667 to 1-667
recover recover file *ex* command 1-667 to 1-667
 red command 1-195 to 1-206
 redirection 1-212 to 1-536
 redraw screen *vi* command, **z** 1-649 to 1-649
 redraw screen *vi* command, **CTRL-L** 1-657 to 1-657
 references 3-27 to 3-28
 regcomp function 3-31 to 3-37
 regerror 1-141 to 1-77, 3-29 to 3-31
 regerror function 1-261 to 1-261, 3-31 to 3-31
 regexec function 3-31 to 3-31
 regexp 1-139 to 1-690, 3-31 to 3-37
 regfree function 3-31 to 3-31
 registers, *dc* 1-155 to 1-155
 regular expressions 1-196 to 1-661, 3-32 to 3-32
 regular expressions, class 1-662 to 1-662
 reject file 1-409 to 1-409
 REJECT *lex* statement 3-8 to 3-8
 released state 1-458 to 1-458
 remainder, *dc* 1-158 to 1-158
 remap *let* macros contain macros *vi* variable 1-665 to 1-675
 remove directory 1-497 to 1-497
 remove files 1-494 to 1-494
 removing attributes 1-639 to 1-639
 removing duplicate lines 1-634 to 1-634
 renice command 1-392 to 1-485
 repeat character search *vi* command, **;** 1-653 to 1-653
 repeat last command *vi* command, **.** 1-657 to 1-657
 repeat last insertion *vi* command, **CTRL-@** 1-659 to 1-659
 repeat last substitute *ex* command, **&** 1-671 to 1-671
 repeat last substitute *ex* command, **~** 1-671 to 1-671
 repeat last substitute *vi* command, **&** 1-657 to 1-657
 repeat string search *vi* command, **n** 1-652 to 1-652
 repetitions of expression metacharacter, ***** 1-661 to 1-661
 replace character *vi* command, **r** 1-656 to 1-656
 replace string *ex* command, **substitute** 1-668 to 1-668
 replacement from last substitute, **~** 1-662 to 1-662
 REPLY environment variable 1-478 to 1-544
 Replyall environment variable 1-322 to 1-324
 replyall environment variable 1-333 to 1-333
 report major changes *vi* variable, *report* 1-675 to 1-675
 report report major changes *vi* variable 1-675 to 1-675
 reserved word commands, shell 1-523 to 1-523
 restart file list *ex* command, **rewind** 1-667 to 1-667
 restrict restricted processing *vi* variable 1-675 to 1-675
 restricted processing *vi* variable, *restrict* 1-675 to 1-675
 restricted shell 1-521 to 1-521
 return command 1-214 to 1-486

rev command 1-487 to 1-487
 reverse character search *vi* command, **,** 1-653 to 1-653
 reverse lines 1-487 to 1-487
 Reverse Polish notation 1-155 to 1-155
 reverse sort 1-560 to 1-560
 reverse string search *vi* command, **N** 1-652 to 1-652
 revision 1-271 to 1-95
 Revision keyword 1-113 to 1-271
 revision name, symbolic 1-111 to 1-97
 revision number 1-111 to 1-95
 revision range 1-457 to 1-489
 revision, checked-out 1-474 to 1-474
 revision, compare RCS 1-470 to 1-470
 revision, head 1-488 to 1-489
 revision, initial 1-95 to 1-97
 revision, lock 1-457 to 1-490
 revision, outdated 1-458 to 1-458
rewind restart file list *ex* command 1-667 to 1-667
 RLENGTH variable 1-26 to 1-26
 rlog command 1-104 to 1-493, 3-26 to 3-26
 rm command 1-132 to 1-496
 rmdir command 1-373 to 1-498, 3-49 to 3-49
 root directory 1-496 to 1-93
 RS variable 1-24 to 1-24
 rsh command 1-519 to 1-521
 rsh file name 1-519 to 1-519
 RSTART variable 1-26 to 1-26
 rules, search 1-526 to 1-698
 run command file *ex* command, **source** 1-668 to 1-678
 run shell *ex* command, **!** 1-670 to 1-670
 rvi command 1-675 to 1-675

S

s change character *vi* command 1-655 to 1-655
S change line *vi* command 1-655 to 1-655
 save environment variable 1-325 to 1-325
 saves buffer for recovery *ex* command, **preserve** 1-666 to 1-666
 scale 1-45 to 1-47
 scale function 1-48 to 1-59
 scale variable 1-47 to 1-61
 scaling factor, *dc* 1-157 to 1-157
 SCCS conversion 1-499 to 1-499
 sccs2rcs command 1-499 to 1-505
 screen environment variable 1-311 to 1-325
 screen size *vi* variable, *window* 1-677 to 1-677
 script, edit 1-175 to 1-506
 script, shell 1-189 to 1-556
 scroll back a line *vi* command, **CTRL-Y** 1-649 to 1-649
 scroll back half screen *vi* command, **CTRL-U** 1-649 to 1-649
 scroll back page *vi* command, **CTRL-B** 1-649 to 1-649
 scroll distance to scroll *vi* variable 1-670 to 1-675
 scroll forward a line *vi* command, **CTRL-E** 1-649 to 1-649
 scroll forward half screen *vi* command, **CTRL-D** 1-649 to 1-649
 scroll forward page *vi* command, **CTRL-F** 1-649 to 1-649
 scrolling commands 1-648 to 1-648

scrolling commands `vi` 1-649 to 1-649
 search backwards for character `vi` command, **F** 1-652 to 1-652
 search backwards for character `vi` command, **T** 1-652 to 1-652
 search backwards for matching string `vi` command, **?** 1-653 to 1-653
 search directory 1-231 to 1-231
 search for files 1-231 to 1-231
 search forward for character `vi` command, **f** 1-652 to 1-652
 search forward for character `vi` command, **t** 1-652 to 1-652
 search forwards for matching string `vi` command, **/** 1-653 to 1-653
 search path 1-526 to 1-698
 search rules 1-526 to 1-698
 searching for strings, `grep` 1-259 to 1-259
 SECONDS environment variable 1-544 to 1-544
 section macro names `vi` variable, `sections` 1-653 to 1-675
 sections section macro names `vi` variable 1-653 to 1-675
 sed command 1-142 to 1-514, 3-32 to 3-37
 segment prefix, program 1-544 to 1-544
 select shell command 1-127 to 1-68
 send mail 1-458 to 1-458
 sendmail environment variable 1-325 to 1-333
 sendwait environment variable 1-325 to 1-333
 separator, internal field 1-543 to 1-543
 server, mail 1-322 to 1-322
 set command 1-10 to 1-518, 3-38 to 3-45
set set variable value `ex` command 1-672 to 1-672
set set variable value `vi` command 1-667 to 1-667
 set variable value `ex` command, **set** 1-672 to 1-672
 set variable value `vi` command, **set** 1-667 to 1-667
 SETMSG command 1-371 to 1-371
 sh command A-19 to A-19, 1-10 to 1-555, 3-28 to 3-45
 sh file name 1-519 to 1-519
 sh prompt string 1-544 to 1-544
 shared file 1-457 to 1-458
 shedit 1-226 to 1-555, 3-38 to 3-45
 shell 1-127 to 1-8
 shell archive 1-541 to 1-541
 shell arrays 1-539 to 1-539
 shell choose command interpreter `ex` variable 1-668 to 1-675
 shell command 1-678 to 1-678
 shell command, `case` 1-523 to 1-523
 shell command, `elif` 1-524 to 1-524
 shell command, `else` 1-524 to 1-524
 shell command, `for` 1-127 to 1-68
 shell command, `function` 1-524 to 1-524
 shell command, `if` 1-521 to 1-615
 shell command, `in` 1-524 to 1-524
 shell command, `select` 1-127 to 1-68
 shell command, `then` 1-524 to 1-524
 shell command, `until` 1-127 to 1-68
 shell command, `while` 1-127 to 1-68
 shell commands 1-521 to 1-521
 shell comments 1-521 to 1-521
 SHELL environment variable 1-197 to 1-678, 3-4 to 3-4
 shell escape `ed` command 1-201 to 1-201
 shell execution environment 1-539 to 1-539
 shell files 1-44 to 1-44
 shell function 1-486 to 1-621
shell invoke subshell `ex` command 1-668 to 1-668
 SHELL macro 1-353 to 1-359
 shell operator, **&** 1-522 to 1-522
 shell operator, **|&** 1-523 to 1-523
 shell reserved word commands 1-523 to 1-523
 shell script 1-189 to 1-556
 shell syntax 1-521 to 1-521
 shell variables 1-539 to 1-639
 shell, interactive 1-520 to 1-520
 shell, login 1-224 to 1-521
 shell, privileged 1-545 to 1-545
 shell, restricted 1-521 to 1-521
 SHELLMETAS environment variable 1-348 to 1-348
 shift command 1-555 to 1-556
 shift left `ex` command, **<** 1-670 to 1-670
 shift left `vi` command, **<** 1-655 to 1-655
 shift positional parameters 1-556 to 1-556
 shift right `ex` command, **>** 1-670 to 1-670
 shift right `vi` command, **>** 1-655 to 1-655
 shiftwidth indentation distance `vi` variable 1-655 to 1-676
 short circuit evaluation 1-22 to 1-22
 short type 2-4 to 2-4
 show matching brackets `vi` variable, `showmatch` 1-676 to 1-676
 SHOWJOB command 1-701 to 1-701
 showmatch show matching brackets `vi` variable 1-676 to 1-676
 showto environment variable 1-325 to 1-325
 Sign environment variable 1-316 to 1-325
 sign environment variable 1-316 to 1-325
 signal.h include file A-14 to A-14
 signals 1-280 to 1-615
 size, block 1-397 to 1-397
 sleep command 1-557 to 1-558
 software distribution 1-97 to 1-97
 sort command 1-120 to 1-566, 2-7 to 2-7, 3-2 to 3-2
 SORT command 1-566 to 1-566
 sort, dictionary 1-559 to 1-559
 sort, month 1-559 to 1-559
 sort, reverse 1-560 to 1-560
 sorted files 1-119 to 1-119
 sorted files, compare 1-119 to 1-119
 sortgen command 1-566 to 1-566
 sorting files 1-559 to 1-559
 sorting key 1-561 to 1-561
 source 1-271 to 1-271
 source code, C 1-143 to 1-143
 source files, C 1-227 to 1-227
 Source keyword 1-113 to 1-271
source run command file `ex` command 1-668 to 1-678
 space move right `vi` command 1-650 to 1-650
 space, disk 1-191 to 1-191
 spaces to tabs 1-632 to 1-632

special built-in commands 1-540 to 1-540
 special characters, quoting 1-528 to 1-528
 special file 1-376 to 1-376
 split command 1-139 to 1-569
 split output stream 1-596 to 1-596
 splitting files 1-138 to 1-567
 sprintf function 1-39 to 1-39
 square root, dc 1-159 to 1-159
 ssize_t type 1-249 to 1-249
 stable state 1-458 to 1-458
 stack 1-156 to 1-156
 standard error 1-707 to 1-707
 standard input 1-174 to 1-98
 standard input, closing 1-537 to 1-537
 standard output 1-111 to 1-5
 standard output, closing 1-537 to 1-537
 start of line metacharacter, ^ 1-661 to 1-661
 start of word metacharacter, < 1-661 to 1-661
 startup.mk file name 1-342 to 1-359
 stat function 2-9 to 2-9
 state 1-109 to 1-98
 State keyword 1-113 to 1-271
 state, experimental 1-458 to 1-98
 state, released 1-458 to 1-458
 state, stable 1-458 to 1-458
 statements, awk 1-29 to 1-29
 states.out file name 1-711 to 1-711
 status, exit 1-118 to 1-617
 stop edit ex command, **suspend** 1-668 to 1-668
 stop editor ex command, **stop** 1-668 to 1-668
 stop editor vi command, **CTRL-Z** 1-658 to 1-658
stop stop editor ex command 1-668 to 1-668
 stream editor 1-506 to 1-506
 strict lock 1-457 to 1-98
 strict locking 1-113 to 1-95
 string type 2-4 to 2-4
 string, ed prompt 1-195 to 1-195
 string, sh prompt 1-544 to 1-544
 strings command 1-570 to 1-571
 strings, dc executing 1-161 to 1-161
 strings, grep matching 1-259 to 1-259
 strings, grep searching for 1-259 to 1-259
 strip command 1-572 to 1-574
 stty command A-19 to A-19, A-19 to A-19, 1-575 to 1-581
 SUBSEP variable 1-19 to 1-30
 subshell 1-522 to 1-9
 subshell environment 1-539 to 1-539
 substitute ed command 1-199 to 1-199
substitute replace string ex command 1-668 to 1-668
 substitution, command 1-534 to 1-534
 substitution, directory 1-529 to 1-529
 substitution, parameter 1-529 to 1-529
 subtraction, dc 1-158 to 1-158
 suffix, file name 1-44 to 1-456
 sum command 1-105 to 1-583
 support, multiple volume 1-135 to 1-421
 suppress diagnostic output 1-467 to 1-470
suspend stop edit ex command 1-668 to 1-668

swapping bytes 1-135 to 1-169
 swapping words 1-135 to 1-135
 switch command 1-523 to 1-523
 symbolic name 1-113 to 1-113
 symbolic revision name 1-111 to 1-97
 symlink function 1-427 to 1-427
 SYMTAB array 1-21 to 1-21
 syntax, shell 1-521 to 1-521
 syserr A-16 to A-16
 syserror A-10 to A-10, A-11 to A-11, A-11 to A-11, A-13 to A-13, A-17 to A-17, A-6 to A-6, A-6 to A-6, A-7 to A-7, A-8 to A-8, A-8 to A-8, A-8 to A-8, 1-100 to 1-99, 3-11 to 3-50
 system files 1-598 to 1-598
 system, awk 1-28 to 1-28

T

t copy text ex command 1-668 to 1-668
 T search backwards for character vi command 1-652 to 1-652
 t search forward for character vi command 1-652 to 1-652
 tab expansion 1-215 to 1-215
 tab stop distances vi variable, **tabstop** 1-672 to 1-676
 tabs 1-632 to 1-632
 tabs command 1-584 to 1-587
 tabs, spaces to 1-632 to 1-632
tabstop tab stop distances vi variable 1-672 to 1-676
 tag name ex command, **tag** 1-668 to 1-668
tag tag name ex command 1-668 to 1-668
 tag vi command, **CTRL-]** 1-653 to 1-653
 taglength length of tags vi variable 1-669 to 1-676
 tags 1-647 to 1-676
 tags file format 1-143 to 1-146, 2-7 to 2-7
 tags file name 1-143 to 1-668, 2-7 to 2-7
 tags name tags file vi variable 1-668 to 1-676
 tail command 1-264 to 1-589
 talk command 1-370 to 1-704
 tape archive 1-590 to 1-590
 tar command A-19 to A-19, 1-126 to 1-595, 2-10 to 2-9
 tar file format 1-137 to 1-595, 2-3 to 2-10
 tcgetattr function 1-580 to 1-580
 tcsetattr function 1-580 to 1-580
 tee command 1-596 to 1-597
 TELL command 1-371 to 1-704
 temporary directory vi variable, **directory** 1-673 to 1-673
 temporary files 1-113 to 1-99
 temporary files, vi 1-678 to 1-678
 TERM environment variable 1-325 to 1-678, 3-4 to 3-4
 term terminal type vi variable 1-676 to 1-676
 terminal name 1-618 to 1-618
 terminal type vi variable, **term** 1-676 to 1-676
 terminate process 1-280 to 1-280
 terminating the shell 1-214 to 1-214
 TERMINFO environment variable 1-201 to 1-679
 terse messages vi variable, **terse** 1-676 to 1-676
terse terse messages vi variable 1-676 to 1-676
 test command 1-222 to 1-602

test condition 1-598 to 1-598
 test conditional expression 1-598 to 1-598
 text editor 1-174 to 1-506
 text files 1-174 to 1-98
 text formatting 1-238 to 1-238
 text insertion commands, **vi** 1-648 to 1-648
 text, descriptive 1-456 to 1-98
 then shell command 1-524 to 1-524
 tilde 1-529 to 1-529
 time 1-109 to 1-110
 time, access 1-128 to 1-300
 time, change 1-232 to 1-232
 time, local 1-110 to 1-110
 time, modification 1-128 to 1-299
 timezone 1-153 to 1-154, 3-4 to 3-53
 TMOUT environment variable 1-544 to 1-601
 TMP environment variable 1-678 to 1-679
 TMPDIR environment variable 1-126 to 1-73, 3-14 to 3-4
 tobyte command A-3 to A-3, 1-244 to 1-604
 toggle case of character **vi** command, **~** 1-657 to 1-657
 token 1-661 to 1-661
 toplines environment variable 1-314 to 1-325
 touch command 1-154 to 1-607
 tput command 1-608 to 1-610
 tr command 1-173 to 1-614
 tracked alias 1-262 to 1-9
 translate characters 1-611 to 1-611
 transmission, file 1-641 to 1-644
 trap command 1-189 to 1-616
 trap interrupts 1-615 to 1-615
 troff program 1-175 to 1-675
 true command 1-118 to 1-617
 trunk 1-109 to 1-98, 3-25 to 3-25
 tsmail 1-333 to 1-333, 3-54 to 3-55
 tsmail command 1-333 to 1-333, 3-54 to 3-55
 tty command A-19 to A-19, A-19 to A-19, 1-618 to 1-619
 turn on talk and write permissions **vi** variable, **mesg**
 1-674 to 1-674
 type command 1-550 to 1-620
 type over characters **vi** command, **R** 1-656 to 1-656
 type, byte 2-4 to 2-4
 type, device 1-376 to 1-376
 type, file 1-227 to 1-234
 type, long 2-4 to 2-4
 type, short 2-4 to 2-4
 type, **ssize_t** 1-249 to 1-249
 type, string 2-4 to 2-4
 typeset command 1-10 to 1-623
 TZ environment variable 1-153 to 1-606, 3-4 to 3-53

U

U undo changes to current line **vi** command 1-657 to
 1-657
u undo last change **vi** command 1-657 to 1-657
 ulimit command 1-555 to 1-555
 umask 1-624 to 1-624
 umask command 1-373 to 1-625
unabbreviate delete abbreviation **ex** command 1-669 to
 1-669

unalias command 1-10 to 1-626
uname command 1-149 to 1-628
uncompress command 1-103 to 1-631
undo change ed command 1-200 to 1-200
undo change RCS 1-475 to 1-475
undo changes to current line vi command, **U** 1-657 to
 1-657
undo last change ex command, **undo** 1-669 to 1-669
undo last change vi command, **u** 1-657 to 1-657
undo undo last change ex command 1-669 to 1-669
unexpand command 1-216 to 1-633
uniq command 1-108 to 1-636
 unique lines 1-634 to 1-634
 UNIX 1-109 to 1-95
unlock 1-109 to 1-466
unmap delete macro **ex** command 1-669 to 1-669
 unnamed buffer 1-654 to 1-654
unpack command 1-126 to 1-638
 unprintable characters 1-81 to 1-81
 unput function 3-7 to 3-7
unset command 1-555 to 1-640
until shell command 1-127 to 1-68
 uppercase conversion 1-170 to 1-170
 usage, disk 1-191 to 1-191
 use pattern matching metacharacters **vi** variable, **magic**
 1-661 to 1-674
 user ID 1-92 to 1-92
 user name 1-269 to 1-269
 USTAR 1-590 to 1-590
uudecode command 1-641 to 1-643
uuencode command 1-641 to 1-645

V

v global on non-matching lines **ex** command 1-669 to
 1-669
VAR environment variable 1-118 to 1-118
 variable attributes 1-621 to 1-621
 variable prefix, **yacc** 1-711 to 1-711
 variable, to fixed record conversion 1-169 to 1-169
 variable, **\$STDLIST** 1-693 to 1-704
 variable, **allnet** environment 1-320 to 1-333
 variable, **append** environment 1-321 to 1-321
 variable, **ARGC** 1-23 to 1-24
 variable, **ask** environment 1-321 to 1-321
 variable, **askbcc** environment 1-321 to 1-321
 variable, **askccc** environment 1-321 to 1-321
 variable, **asksub** environment 1-321 to 1-321
 variable, **autoprint** environment 1-307 to 1-321
 variable, **bang** environment 1-318 to 1-321
 variable, **CDPATH** environment 1-542 to 1-84
 variable, **cmd** environment 1-312 to 1-332
 variable, **COLUMNS** environment 1-201 to 1-69, 3-3 to 3-38
 variable, **configuration** 1-247 to 1-247
 variable, **conv** environment 1-305 to 1-333
 variable, **CONVFMT** 1-19 to 1-19
 variable, **crt** environment 1-313 to 1-322
 variable, **DEAD** environment 1-316 to 1-325
 variable, **dot** environment 1-322 to 1-322

variable, ECHO environment 1-14 to 1-75
 variable, EDITOR environment 1-310 to 1-545, 3-10 to 3-9
 variable, ENV environment 1-10 to 1-678, 3-3 to 3-3
 variable, environment 1-515 to 1-515
 variable, ERRNO environment 1-542 to 1-554
 variable, escape environment 1-316 to 1-332
 variable, EXINIT environment 1-647 to 1-678
 variable, FCEDIT environment 1-225 to 1-542
 variable, flipr environment 1-313 to 1-324
 variable, FNR 1-25 to 1-25
 variable, folder environment 1-309 to 1-330
 variable, FS 1-18 to 1-27
 variable, header environment 1-323 to 1-323
 variable, HELP environment 1-265 to 1-265
 variable, HELPCMD environment 1-265 to 1-265
 variable, HISTFILE environment 1-224 to 1-548
 variable, HISTSIZE environment 1-224 to 1-601
 variable, hold environment 1-307 to 1-323
 variable, HOME environment A-2 to A-2, A-2 to A-2, 1-201 to 1-84, 3-3 to 3-3
 variable, HPLDEVNO 1-619 to 1-619
 variable, HPQUIET 1-371 to 1-371
 variable, ibase 1-48 to 1-66
 variable, IFS environment 1-478 to 1-543
 variable, ignore environment 1-305 to 1-323
 variable, ignoreeof environment 1-322 to 1-323
 variable, indent environment 1-323 to 1-323
 variable, indentprefix environment 1-317 to 1-323
 variable, keep environment 1-323 to 1-323
 variable, keepsave environment 1-306 to 1-323
 variable, LC_COLLATE environment 1-249 to 1-612, 3-30 to 3-30
 variable, LC_CTYPE environment 1-612 to 1-613, 3-30 to 3-30
 variable, LC_MESSAGES environment 1-233 to 1-494
 variable, LC_TIME environment 1-152 to 1-152
 variable, LINE environment 1-384 to 1-384
 variable, LINENO environment 1-543 to 1-554
 variable, LINES environment 1-382 to 1-678, 3-3 to 3-3
 variable, LINE_MAX 1-261 to 1-261
 variable, LISTER environment 1-323 to 1-323
 variable, LOGNAME environment 1-298 to 1-320, 3-3 to 3-3
 variable, MAIL environment 1-543 to 1-543
 variable, MAILCHECK environment 1-543 to 1-601
 variable, MAILDIR environment 1-320 to 1-326
 variable, MAILER environment 1-458 to 1-77, 3-3 to 3-3
 variable, MAILPATH environment 1-527 to 1-543
 variable, MAILRC environment 1-319 to 1-332
 variable, MAKEFLAGS environment 1-350 to 1-355
 variable, MAKESTARTUP environment 1-335 to 1-353
 variable, MANPAGER environment 1-363 to 1-365
 variable, MANPATH environment 1-362 to 1-365
 variable, MBOX environment 1-323 to 1-326
 variable, msg 1-674 to 1-674
 variable, metoo environment 1-324 to 1-324
 variable, MORE environment 1-382 to 1-382
 variable, NF 1-25 to 1-25
 variable, noallnet environment 1-321 to 1-321
 variable, noappend environment 1-321 to 1-321
 variable, noask environment 1-321 to 1-321
 variable, noaskbcc environment 1-321 to 1-321
 variable, noaskcc environment 1-321 to 1-321
 variable, noasksub environment 1-321 to 1-321
 variable, noautoprint environment 1-320 to 1-321
 variable, nobang environment 1-321 to 1-321
 variable, nocrt environment 1-322 to 1-322
 variable, nodot environment 1-322 to 1-322
 variable, noflipr environment 1-322 to 1-322
 variable, nohold environment 1-323 to 1-323
 variable, noignore environment 1-323 to 1-323
 variable, noignoreeof environment 1-323 to 1-323
 variable, nokeep environment 1-323 to 1-323
 variable, nokeepsave environment 1-323 to 1-323
 variable, nometoo environment 1-324 to 1-324
 variable, nooutfolder environment 1-324 to 1-324
 variable, nopage environment 1-324 to 1-324
 variable, noquiet environment 1-324 to 1-324
 variable, norecord environment 1-324 to 1-324
 variable, nosave environment 1-325 to 1-325
 variable, nosendwait environment 1-325 to 1-325
 variable, noshowto environment 1-325 to 1-325
 variable, noSign environment 1-325 to 1-325
 variable, noSIGN environment 1-325 to 1-325
 variable, NR 1-25 to 1-25
 variable, obase 1-48 to 1-49
 variable, OFMT 1-18 to 1-31
 variable, OFS 1-31 to 1-31
 variable, OLDPWD environment 1-529 to 1-83
 variable, onehop environment 1-324 to 1-333
 variable, OPTARG 1-256 to 1-257
 variable, OPTIND 1-256 to 1-257
 variable, OPTIND environment 1-546 to 1-601
 variable, ORS 1-31 to 1-31
 variable, out environment 1-537 to 1-537
 variable, outfolder environment 1-324 to 1-324
 variable, page environment 1-312 to 1-324
 variable, PAGER environment 1-313 to 1-364
 variable, PATH environment 1-121 to 1-9, 3-3 to 3-3
 variable, PID environment 1-544 to 1-544
 variable, prompt environment 1-324 to 1-324
 variable, PS1 environment 1-527 to 1-544
 variable, PS2 environment 1-479 to 1-544
 variable, PS3 environment 1-525 to 1-554
 variable, PS4 environment 1-527 to 1-544
 variable, PWD environment 1-516 to 1-83
 variable, quiet environment 1-324 to 1-324
 variable, RANDOM environment 1-544 to 1-544
 variable, RCSPATH environment 1-103 to 1-501, 3-12 to 3-15
 variable, record environment 1-304 to 1-324
 variable, REPLY environment 1-478 to 1-544
 variable, Replyall environment 1-322 to 1-324
 variable, replyall environment 1-333 to 1-333
 variable, RLENGTH 1-26 to 1-26
 variable, RS 1-24 to 1-24
 variable, RSTART 1-26 to 1-26
 variable, save environment 1-325 to 1-325
 variable, scale 1-47 to 1-61

Index-22

variable, screen environment 1-311 to 1-325
 variable, SECONDS environment 1-544 to 1-544
 variable, sendmail environment 1-325 to 1-333
 variable, sendwait environment 1-325 to 1-333
 variable, SHELL environment 1-197 to 1-678, 3-4 to 3-4
 variable, SHELLMETAS environment 1-348 to 1-348
 variable, showto environment 1-325 to 1-325
 variable, Sign environment 1-316 to 1-325
 variable, sign environment 1-316 to 1-325
 variable, SUBSEP 1-19 to 1-30
 variable, TERM environment 1-325 to 1-678, 3-4 to 3-4
 variable, TERMINFO environment 1-201 to 1-679
 variable, TMOU environment 1-544 to 1-601
 variable, TMP environment 1-678 to 1-679
 variable, TMPDIR environment 1-126 to 1-73, 3-14 to 3-4
 variable, toplines environment 1-314 to 1-325
 variable, TZ environment 1-153 to 1-606, 3-4 to 3-53
 variable, VAR environment 1-118 to 1-118
 variable, VISUAL environment 1-315 to 1-545, 3-10 to 3-9
 variable, window 1-211 to 1-211
 variable, wrapscan 1-677 to 1-677
 variable, yyin 3-6 to 3-6
 variable, yyleng 3-6 to 3-6
 variable, yylineno 3-6 to 3-7
 variable, yyout 3-6 to 3-6
 variables 1-285 to 1-481
 variables, awk 1-19 to 1-19
 variables, display 1-621 to 1-621
 variables, local 1-621 to 1-621
 variables, shell 1-539 to 1-639
version editor version ex command 1-669 to 1-669
 vfprintf function 3-7 to 3-7
 vi absolute movement 1-648 to 1-650
 vi command 1-143 to 1-690, 2-7 to 2-7, 3-32 to 3-9, 7
 vi command, ! filter 1-655 to 1-655
 vi command, \$ move to end of line 1-651 to 1-651
 vi command, % move to balancing bracket 1-653 to 1-653
 vi command, & repeat last substitute 1-657 to 1-657
 vi command, (move to previous sentence 1-652 to 1-652
 vi command,) move to next sentence 1-652 to 1-652
 vi command, + move to start of next line 1-650 to 1-650
 vi command, , reverse character search 1-653 to 1-653
 vi command, . repeat last command 1-657 to 1-657
 vi command, / search forwards for matching string 1-653 to 1-653
 vi command, O move to start of line 1-650 to 1-650
 vi command, : invoke ex command from 1-657 to 1-657
 vi command, ; repeat character search 1-653 to 1-653
 vi command, < shift left 1-655 to 1-655
 vi command, > shift right 1-655 to 1-655
 vi command, ? search backwards for matching string 1-653 to 1-653
 vi command, @ delete inserted line 1-659 to 1-659
 vi command, @ invoke macro 1-657 to 1-657
 vi command, A append at end of line 1-656 to 1-656
 vi command, a append text 1-656 to 1-656
 vi command, B move to previous full word 1-651 to 1-651
 vi command, b move to previous word 1-651 to 1-651
 vi command, backspace delete last character 1-658 to 1-658
 vi command, backspace move left 1-650 to 1-650
 vi command, c change 1-654 to 1-654
 vi command, C change to end of line 1-655 to 1-655
 vi command, CTRL-j tag 1-653 to 1-653
 vi command, d delete 1-655 to 1-655
 vi command, D delete to end of line 1-655 to 1-655
 vi command, E move to end of full word 1-651 to 1-651
 vi command, e move to end of word 1-652 to 1-652
 vi command, ESC leave insert mode 1-659 to 1-659
 vi command, F search backwards for character 1-652 to 1-652
 vi command, f search forward for character 1-652 to 1-652
 vi command, G move to absolute line 1-650 to 1-650
 vi command, h move left 1-650 to 1-650
 vi command, H move to top of screen 1-652 to 1-652
 vi command, I insert at start of line 1-656 to 1-656
 vi command, i insert text 1-656 to 1-656
 vi command, J join lines 1-656 to 1-656
 vi command, j move down one line 1-650 to 1-650
 vi command, k move up one line 1-650 to 1-650
 vi command, l move right 1-650 to 1-650
 vi command, L move to bottom of screen 1-652 to 1-652
 vi command, m mark current position 1-650 to 1-650
 vi command, M move to middle of screen 1-652 to 1-652
 vi command, map define macro 1-665 to 1-665
 vi command, mark name 1-650 to 1-650
 vi command, n repeat string search 1-652 to 1-652
 vi command, N reverse string search 1-652 to 1-652
 vi command, o open new line after current 1-656 to 1-656
 vi command, O open new line before current 1-656 to 1-656
 vi command, p paste after 1-657 to 1-657
 vi command, P paste before 1-656 to 1-656
 vi command, Q enter ex mode 1-657 to 1-657
 vi command, r replace character 1-656 to 1-656
 vi command, R type over characters 1-656 to 1-656
 vi command, s change character 1-655 to 1-655
 vi command, S change line 1-655 to 1-655
 vi command, set set variable value 1-667 to 1-667
 vi command, space move right 1-650 to 1-650
 vi command, T search backwards for character 1-652 to 1-652
 vi command, t search forward for character 1-652 to 1-652
 vi command, U undo changes to current line 1-657 to 1-657
 vi command, u undo last change 1-657 to 1-657
 vi command, W move to next full word 1-652 to 1-652
 vi command, w move to next word 1-652 to 1-652
 vi command, x delete character 1-655 to 1-655
 vi command, X delete previous character 1-655 to 1-655
 vi command, Y yank line 1-655 to 1-655
 vi command, y yank text to buffer 1-655 to 1-655
 vi command, z redraw screen 1-649 to 1-649
 vi command, ZZ write and exit 1-657 to 1-657
 vi command, [[move back to start of section 1-653 to 1-653
 vi command, CTRL-@ repeat last insertion 1-659 to 1-659
 vi command, CTRL-B scroll back page 1-649 to 1-649
 vi command, CTRL-D decrement autoindent 1-658 to 1-658

- vi command, **CTRL-D** scroll forward half screen 1-649 to 1-649
- vi command, **CTRL-E** scroll forward a line 1-649 to 1-649
- vi command, **CTRL-F** scroll forward page 1-649 to 1-649
- vi command, **CTRL-G** file information 1-657 to 1-657
- vi command, **CTRL-H** delete last character 1-658 to 1-658
- vi command, **CTRL-H** move left 1-650 to 1-650
- vi command, **CTRL-J** move down one line 1-650 to 1-650
- vi command, **CTRL-L** redraw screen 1-657 to 1-657
- vi command, **CTRL-M** move to start of next line 1-650 to 1-650
- vi command, **CTRL-N** move down one line 1-650 to 1-650
- vi command, **CTRL-P** move up one line 1-650 to 1-650
- vi command, **CTRL-Q** insert character literally 1-658 to 1-658
- vi command, **CTRL-T** increment autoindent 1-658 to 1-658
- vi command, **CTRL-U** scroll back half screen 1-649 to 1-649
- vi command, **CTRL-V** insert character literally 1-658 to 1-678
- vi command, **CTRL-W** back up a word 1-659 to 1-659
- vi command, **CTRL-Y** scroll back a line 1-649 to 1-649
- vi command, **CTRL-Z** stop editor 1-658 to 1-658
- vi command, **CTRL-^** edit alternate file 1-658 to 1-658
- vi command, **^** move to start of marked line 1-651 to 1-651
- vi command, **`** move to mark 1-651 to 1-651
- vi command, **-** move to start of previous line 1-651 to 1-651
- vi command, **]** move to next section 1-653 to 1-653
- vi command, **^** move to first non-blank 1-651 to 1-651
- vi command, **{** move back to start of paragraph 1-652 to 1-652
- vi command, **|** move to column 1-651 to 1-651
- vi command, **}** move to next paragraph 1-652 to 1-652
- vi command, **~** toggle case of character 1-657 to 1-657
- vi context-dependent movement 1-648 to 1-651
- vi manipulation commands 1-648 to 1-648
- vi mode 1-646 to 1-657
- vi movement commands 1-648 to 1-648
- vi temporary files 1-678 to 1-678
- vi text insertion commands 1-648 to 1-648
- vi variable, **autoindent** automatic indent 1-658 to 1-676
- vi variable, **autoprint** display current line after change 1-672 to 1-672
- vi variable, **autowrite** 1-666 to 1-673
- vi variable, **beautify** discard control characters 1-673 to 1-673
- vi variable, **directory** temporary directory 1-673 to 1-673
- vi variable, **edcompatible** do ed like substitutions 1-673 to 1-673
- vi variable, **errorbells** precedes error message with alert 1-673 to 1-673
- vi variable, **exrc** ignore .exrc file 1-673 to 1-673
- vi variable, **ignorecase** ignore case in pattern matching 1-673 to 1-673
- vi variable, **list** display lines in verbose form 1-674 to 1-674
- vi variable, **magic** use pattern matching metacharacters 1-661 to 1-674

- vi variable, **maxbuffers** buffer size 1-674 to 1-674
- vi variable, **mesg** turn on talk and write permissions 1-674 to 1-674
- vi variable, **number** display lines with line numbers 1-674 to 1-674
- vi variable, **paragraphs** paragraph macro names 1-652 to 1-674
- vi variable, **quiet** quiet mode 1-675 to 1-675
- vi variable, **readonly** do not write on file 1-647 to 1-675
- vi variable, **remap** let macros contain macros 1-665 to 1-675
- vi variable, **report** report major changes 1-675 to 1-675
- vi variable, **restrict** restricted processing 1-675 to 1-675
- vi variable, **scroll** distance to scroll 1-670 to 1-675
- vi variable, **sections** section macro names 1-653 to 1-675
- vi variable, **shiftwidth** indentation distance 1-655 to 1-676
- vi variable, **showmatch** show matching brackets 1-676 to 1-676
- vi variable, **tabstop** tab stop distances 1-672 to 1-676
- vi variable, **taglength** length of tags 1-669 to 1-676
- vi variable, **tags** name tags file 1-668 to 1-676
- vi variable, **term** terminal type 1-676 to 1-676
- vi variable, **terse** terse messages 1-676 to 1-676
- vi variable, **warn** warn about modifications 1-677 to 1-677
- vi variable, **window** screen size 1-677 to 1-677
- vi variable, **wrapmargin** wrap around at end of line 1-677 to 1-677
- vi variable, **wrapscreen** wrap around searches 1-677 to 1-677
- vi variable, **writeany** do not write to existing file 1-677 to 1-677
- vi, insert mode 1-659 to 1-659
- vi, object manipulation 1-653 to 1-653
- vi, scrolling commands 1-649 to 1-649
- visual** enter vi mode ex command 1-669 to 1-669
- VISUAL** environment variable 1-315 to 1-545, 3-10 to 3-9
- volume support, multiple 1-135 to 1-421

W

- W** move to next full word vi command 1-652 to 1-652
- w** move to next word vi command 1-652 to 1-652
- wait 1-557 to 1-557
- wait command 1-275 to 1-691
- wall command 1-371 to 1-693
- warn about modifications vi variable, **warn** 1-677 to 1-677
- warn warn about modifications vi variable 1-677 to 1-677
- wc command 1-106 to 1-695
- what is file name 1-364 to 1-365
- whence command 1-555 to 1-697
- which command 1-696 to 1-699
- while shell command 1-127 to 1-68
- white space 1-174 to 1-174, 3-24 to 3-24
- white space, ignore 1-174 to 1-174

who command 1-443 to 1-701
 width, line 1-238 to 1-239
 width, page 1-69 to 1-69
 wild card characters 1-538 to 1-538
 window screen size vi variable 1-677 to 1-677
 window variable 1-211 to 1-211
wn write and edit next ex command 1-669 to 1-669
 word 1-651 to 1-651
 word commands, shell reserved 1-523 to 1-523
 wordexp function 1-550 to 1-552
 words, count 1-694 to 1-694
 words, swapping 1-135 to 1-135
 working directory 1-451 to 1-83
 working directory, previous 1-83 to 1-83
 working file 1-109 to 1-99
 working files, cleanup 1-466 to 1-466
wq write and quit ex command 1-670 to 1-670
 wrap around at end of line vi variable, wrapmargin
 1-677 to 1-677
 wrap around searches vi variable, wrapscan 1-677 to
 1-677
 wrapmargin command 1-678 to 1-678
 wrapmargin wrap around at end of line vi variable
 1-677 to 1-677
 wrapscan variable 1-677 to 1-677
 wrapscan wrap around searches vi variable 1-677 to
 1-677
 write and edit next ex command, **wn** 1-669 to 1-669
 write and exit vi command, **ZZ** 1-657 to 1-657
 write and quit ex command, **wq** 1-670 to 1-670
 write and quit ex command, **xit** 1-670 to 1-670
 write command 1-370 to 1-704
 write file ed command 1-201 to 1-201
 write file ex command, **write** 1-669 to 1-669
 write message to file, ~w 3-9 to 3-9
write write file ex command 1-669 to 1-669
 writeany do not write to existing file vi variable 1-677
 to 1-677

X

x delete character vi command 1-655 to 1-655
X delete previous character vi command 1-655 to 1-655
 xargs command 1-705 to 1-709
xit write and quit ex command 1-670 to 1-670
 xtrace 1-621 to 1-621

Y

Y yank line vi command 1-655 to 1-655
y yank text to buffer vi command 1-655 to 1-655
 y.output file name 1-711 to 1-711
 y.tab.c file name 1-710 to 1-711
 y.tab.h file name 1-710 to 1-711
 yacc command 1-227 to 1-714, 3-8 to 3-8
 yacc variable prefix 1-711 to 1-711
 yank line vi command, **Y** 1-655 to 1-655
 yank text to buffer ex command, **yank** 1-670 to 1-670
 yank text to buffer vi command, **y** 1-655 to 1-655

yank yank text to buffer ex command 1-670 to 1-670
 yycomment function 3-7 to 3-7
 YYDEBUG 1-711 to 1-711
 yyerror function 3-7 to 3-7
 yygetc function 3-7 to 3-7
 yygetc macro 3-7 to 3-7
 yyin variable 3-6 to 3-6
 yylen variable 3-6 to 3-6
 yyless function 3-8 to 3-8
 yylex function 3-6 to 3-8
 yylineno variable 3-6 to 3-7
 YYLMAX 3-6 to 3-6
 yymapch function 3-7 to 3-7
 yymore function 3-8 to 3-8
 yyout variable 3-6 to 3-6
 yytext array 3-6 to 3-6
 yywrap function 3-7 to 3-7

Z

z command 1-677 to 1-677
z display lines ex command 1-670 to 1-670
z redraw screen vi command 1-649 to 1-649
 zcat command 1-126 to 1-716
ZZ write and exit vi command 1-657 to 1-657