

Windows' Linux Look-alikes:

An analysis under copyright law of Linux projects
that emulate the Windows desktop environment.

Douglas Steele

Please direct comments/corrections to douglas@steele.name

.02 Version

October 15, 2003

This document is intended to be useful to developers of free software that may include the ability to emulate aspects of the graphical environment familiar to users of the Microsoft Windows operating system. It is not a substitute for legal advice from an attorney licensed in your jurisdiction and no attorney/client relationship is formed by your reading it.



This document is licensed under the Attribution/Share Alike Creative Commons [license](http://creativecommons.org/licenses/by-sa/1.0/).¹ No rights are granted -- since they are not mine to grant -- in any logo, graphic design, trademarks or trade names used in this document.

¹ [Http://creativecommons.org/licenses/by-sa/1.0/](http://creativecommons.org/licenses/by-sa/1.0/)

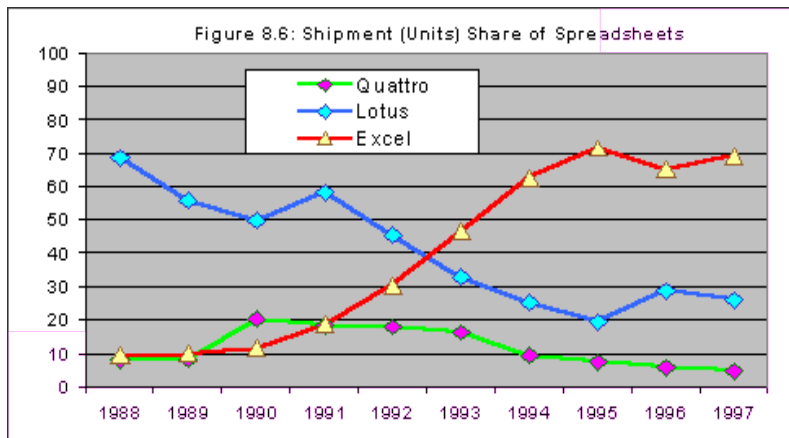
I. Windows through a mirror.

One hundred million new personal computers are sold every year and Microsoft controls more than 90% of the market for their operating systems.² Microsoft “possesses a dominant, persistent, and increasing share of the world- wide market for Intel-compatible PC operating systems³” and has used its power to extend further monopolies into the other markets including browsers (Internet Explorer), word processors (Word), spreadsheets (Excel), and office application suites (Office). It has aggressively entered the server, console game and embedded device markets and acquired significant market share in them.

The only meaningful threat to Microsoft's monopolies is the emergence of “free” and “open source” software projects, primarily the GNU/Linux operating system. Philanthropist Mitch Kapor's Open Source Applications Foundation is predicting that in four years, Linux will be used on up to ten percent of all personal computers and account for a larger share of new shipments.⁴ If Kapor's optimistic prediction about Linux on the desktop proves true, credit will be due partly to his current largess in funding open source programs but even more so to Kapor's greatest failure.

Mitch Kapor was the founder of Lotus Development Corporation and the designer of the Lotus 1-2-3 spreadsheet. In Lotus v. Borland the First Circuit and the Supreme Court, at Kapor's expense, severely limited the copyright privilege afforded to the operation of computers via graphical and menuing interfaces.⁵ The immediate effect of Lotus v. Borland was to allow

Illustration 1 Spreadsheet Market Share 88-97. Lebovitz.



Borland and then Microsoft's Excel to cannibalize Lotus 1-2-3's previously dominant market share.⁶ The favor is being returned by Kapor and tens of thousands open source software developers who are creating alternatives to Microsoft that, relying in part on Lotus v. Borland, mimic the menu structures and look and feel of Microsoft's Windows.

² The Best Is Yet to Come, remarks by Bill Gates to WINHEC 2000, New Orleans, Apr. 25, 2000, <<http://www.microsoft.com/billgates/speeches/04?25winhec00.htm>>.

³ United States v. Microsoft, 65 F. Supp. 2d 1, 1999 WL 1001107 (D.D.C. 1999). Findings of Fact, 2 at 35.

⁴ Bert Decram, Linux on the Desktop Report. Open Source Application Foundation. July 10, 2003. <http://www.osafoundation.org/desktop-linux-overview.pdf>

⁵ Lotus Development Corp. v. Borland International, Inc., 49 F.3d 807 (1st Cir. 1995), aff'd by an equally divided court, 516 U.S. 233 (1996).

⁶ See Stan Lebovitz, *Re-thinking the Network Economy*. New York, NY: Amacom, 2002. <http://www.utdallas.edu/~liebowitz/book/sheets/sheet.html>

A number of Linux based projects offer graphical interfaces or desktop environments that approximate the look and feel of Windows. None, however, does so more explicitly than XPde, a project to develop an open source desktop environment led by Jose Leon, a Spanish programmer that sets out to “recreate the Windows XP interface to-the-pixel point.”⁷ This analysis of U.S. Copyright law as it applies to graphical interfaces and menus that control computer operating systems assesses the risk that projects like XPde run in seeking to mirror elements of the world's richest company's most lucrative products.⁸ To the extent that consistent behavior over many years is predictive of future behavior, the risk is significant.⁹

II. Operating the system

At their most basic physical level, computers are no more complicated than a light bulb. A computer's central processor (CPU) is merely a large collections of circuits that can be toggled quickly. As in a computer, switching a light on or off can be used to store or represent data. Paul Revere's “one if by land, two if by sea” was an early illustration of the principle. The amount of data Revere could store or transmit was constrained both by the number of lamps he could fit in the steeple window (two apparently) and his limitations as a programmer. Barely a decade after Revere's death, Samuel Morse hacked the on/off code idea to produce Morse Code that, using the same lanterns, could have told which way the Redcoats were coming as well as each one's name, rank and serial number.

While computers are conceptually simple — they are just a set of switches after all — the sheer number of toggles makes them difficult to operate. The ENIAC computer developed during World War II used 18,000 vacuum tubes to perform mathematical calculations. The “programmers” had to manually rewire the circuits each time they performed a different calculation. John von Neumann's shared-program concept led to the first abstraction of user controls from the hardware.¹⁰ The first 'graphical' interface on a computer was arguably when the addition of a panel of switches so that circuits could be reprogrammed without rewiring.

In 1945, the ENIAC had 18,000 vacuum tube circuits that was 'programmed' by six women who wired the circuits by hand. In 2003, the CPU alone on an Intel Pentium IV system has 55 million transistorized circuits. Human evolution has not kept pace. In order to control

⁷ XPde Frequently Asked Questions; What is XPde? <http://www.xpde.com/faq.php#1> (XPde FAQ).

⁸ As of July, 2003, Microsoft has \$49 billion dollars in cash on hand and 50,000 employees, of whom 13,000 are developers and 300 are in-house lawyers. XPde has 20 volunteer developers attracted to the project “[b]asically to learn more about Linux application development, and for fun.” See Seattle Post-Intelligencer, “*Microsoft's cash reserve at \$49 billion.*” July 18, 2003; XPde FAQ, supra note 6.

⁹ See e.g. *U.S. v. Microsoft*, Findings of Fact, supra note 2. 6 at 412:

“Through its conduct toward Netscape, IBM, Compaq, Intel and others, Microsoft has demonstrated that it will use its prodigious market power and immense profits to harm any firm that insists on pursuing initiatives that could intensify competition against one of Microsoft's core products.”

¹⁰ See John Von Neumann, “First Draft of a Report on the EDVAC.” 1945. (First written description of a computer program not stored in hardware.)

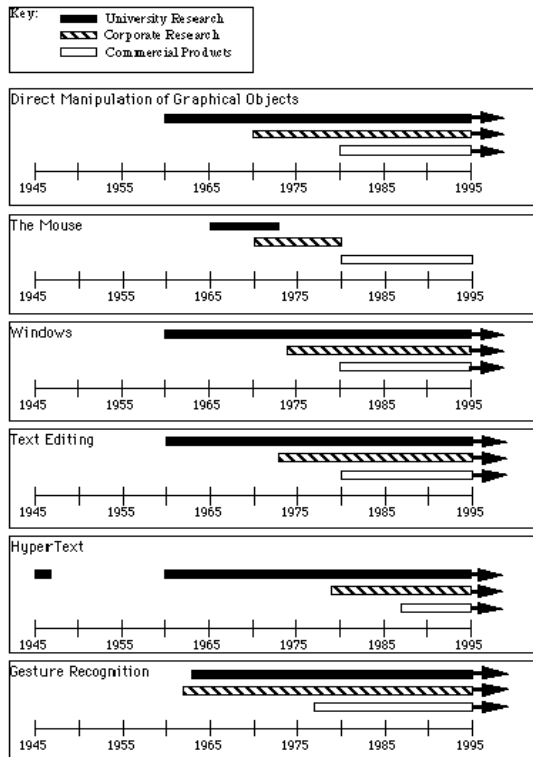


Illustration 2Mayers: Evolution of graphical interface elements

line and typical distributions of Linux weigh in with similar code line numbers.¹¹

As computers grew more complex, programmers developed graphical metaphors to operate the operating systems.¹² The ability to directly manipulate graphical objects enabled metaphors like the desktop and windows combined with physical pointers like the mouse and more powerful graphics enabled hardware, enabled researchers at Xerox's Palo Alto Research Center to fashion the first graphical interface in the 1970s but the idea was borrowed and successfully commercialized by Apple with its Lisa (1983) and Macintosh (1984) computers.

In November 1985, Microsoft released version 1.0 of Windows graphical

these ever more complex machines, we have resorted to simplified interfaces.

Early computers were individually programmed for each user and programs were not typically shared. The invention of the first computer operating system in 1954 by Gene Amdahl allowed multiple programs on the same computer to use a common textual interface as well as to share system software and hardware resources. These changes made it much easier for identical copies of programs to be run on different computers and for users to use different computers without having to learn a unique set of commands for each one.

Software program were developed when computers became too complex to operate directly. Operating systems were then developed as programs became too complex to operate directly. This pattern recurred as operating systems themselves became more complex in the 1970s and 80s. The first operating systems in 1954 shipped with 6,000 lines of code. The Windows XP operating system has 55 million lines of code with 60 characters per line and typical distributions of Linux weigh in with similar code line numbers.¹¹

¹¹ See e.g. Robert Lemos, "Old code in Windows is security threat," June 10, 2002. <http://news.com.com/2100-1001-934363.html>; Alex Leigh, "Open Source," August 28, 2002. <http://www.tessier.com/2001/Life/08/28/opensource/>

Microsoft Windows 3.1 (released in 1992):	3 million lines of code.
Windows 95:	15 million lines.
Windows XP:	50 million lines.

Linux kernel in 1991:	10,000 lines.
Linux kernel with GUI and networking:	170,000 lines.
Current Linux 2.4.20 kernel:	2 million lines.
Complete RedHat 6.2 distribution (2000):	17 million lines.
Redhat 7.1 (2001):	30 million lines.

¹² See e.g. Brad Meyers, *A Brief History of Human Computer Interaction Technology*. ACM interactions. Vol. 5, no. 2, March, 1998. pp. 44-54.

environment which in turn copied the windowing and menuing metaphors used by Apple. Apple and Microsoft initially fought over the scope of graphical interface copyright but Apple then granted a limited license for Microsoft for Windows 1.0. As subsequent releases of Windows versions 2 and 3 grew ever more Maclike in their "look and feel," Apple alleged that Microsoft exceeded its license and infringed Apple's copyright by cloning its graphical user interface. In the litigation that ensued, Apple v. Microsoft,¹³ was successful in borrowing Apple's graphical interface and the case joined Lotus v. Borland as part of the triumvirate of cases that define the scope of copyright privilege for computer operating systems under U.S. Law.¹⁴

Microsoft now alleges that Linux is similarly trying to borrow from Windows and that "[t]here's no question that in cloning activities, IP from many, many companies, including Microsoft, is being used in open-source software."¹⁵ It is with the authority of experience that Chairman Gates goes on to say "[w]hen people clone things, that often becomes unavoidable."

III. Enter the Penguin

The Unix operating system was invented at the Bell Labs in 1969 and was initially licensed by AT&T, along with its source code, to university computer science labs for \$99. AT&T's generous licensing terms stemmed from a settlement of a Department of Justice's antitrust lawsuit launched in 1949 and settled in 1956. The settlement prohibited AT&T from extending its telephone monopoly into other markets such as for example, computer operating systems. With the breakup of AT&T in the early 80s, the settlement order was lifted and licensing fees for Unix quickly reached \$250,000.¹⁶ Even academics and students at universities that could afford licenses lost access to the access to the source code and the ability to learn and tinker.

In Fall 1983, Richard Stallman, a programmer at MIT's Artificial Intelligence Lab sent out an email announcement on a computer discussion list that bordered on the grandiose:

Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed.¹⁷

Stallman and the volunteers he attracted then proceeded to do what he had announced. Or rather, they finished the first 97% by 1991 but were stuck on the hardest part -- the kernel. An operating system's kernel is analogous to an air traffic control tower: the tower is only a small part of an airport, few people ever see the inside of it and things crash if it doesn't work. A solution to the kernel problem came in the form of an improbable July 1991 email

¹³ Apple Computer Inc. v. Microsoft Corporation, 35 F.3d 1435 (9th Cir. 1994).

¹⁴ The third, Computer Assocs. Int'l, Inc. v. Altai, Inc. will be discussed later.

¹⁵ Paula Rooney, "Gates Says SCO's Case Against IBM Will Harm Linux's Commercial Prospects--Contends Microsoft IP likely violated by open source code," CRN, July 25, 2003. Quoting Bill Gates.

¹⁶ Andrew Leonard, "BSD Unix: Power to the people, from the code," Salon May 16, 2001.

¹⁷ Richard Stallman, Original announcement of the GNU Project. September 27, 1983. <http://www.gnu.org/gnu/initial-announcement.html>

announcement on another computer list by an undergraduate at Helsinki University:

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu)...
I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)
Linus (torvalds@kruuna.helsinki.fi)

By Christmas 1991, it was possible to make Linus Torvalds' kernel and the GNU projects' system function as a complete operating system but it was not possible to modify the two to work together and then redistribute the combined whole because of the most salient feature of the GNU system; the General Public License (GPL).¹⁸ The GPL not an end user license of the sort most people who use computers are familiar with.¹⁹ In contrast to proprietary licenses, the GPL allows software released under it to be freely copied, modified, given away or sold. What it requires is that all modified and extended versions of these program be free software as well, a principle known as copyleft. Torvalds released early versions of the Linux kernel under a freeware license of his own devising²⁰ but the release of version 0.12, in January 1992, he adopted the GPL and the first GNU/Linux operating system was complete.²¹

Red Hat in the United States and Suse in Europe are the most used commercial distributions of Linux but more than 300 other 'distros' exist. For those whom even this is a paucity of choices, there is the option to 'roll your own' by using Linux from Scratch to customize the selection of every file in a personalized distribution.²² Distributions of Linux have a few basic commonalities. All run some version of the Linux kernel which is licensed under the GPL. The copyright to the kernel is owned by Linus Torvalds and more than 10,000 other identifiable contributors. All are entitled to use the Linux trademark which is owned by Linus Torvalds, as

¹⁸ See GNU General Public License, Version 2. June 1991. <http://www.gnu.org/copyleft/gpl.html>

¹⁹ The GPL is not an end user license at all. GPLd software can be used freely by anyone. The terms of the GPL only affect modification and redistribution. See GPL §0.

²⁰ The license of the 0.11 kernel states in full:

This kernel is (C) 1991 Linus Torvalds, but all or part of it may be redistributed provided you do the following:

- Full source must be available (and free), if not with the distribution then at least on asking for it.
- Copyright notices must be intact. (In fact, if you distribute only parts of it you may have to add copyrights, as there aren't (C)'s in all files.) Small partial excerpts may be copied without bothering with copyrights.
- You may not distribute this for a fee, not even "handling" costs.

Mail me at "torvalds@kruuna.helsinki.fi" if you have any questions.

Sadly, a kernel by itself gets you nowhere. To get a working system you need a shell, compilers, a library etc. These are separate parts and may be under a stricter (or even looser) copyright. Most of the tools used with linux are GNU software and are under the GNU copyleft. These tools aren't in the distribution - ask me (or GNU) for more info.

²¹ There exists a debate, often theological in tone, over the choice of the name GNU/Linux versus just Linux. In choosing to side with the adherents of GNU/Linux, I find the Torvalds' rationale (supra note 17) to be dispositive: most of the tools used with the Linux kernel are GNU and GPLd.

²² See Linuxfromscratch.org

well as “Tux,” the Penguin trademark. Of the thousands of files in a typical distribution, about 70% are licensed under the GPL and the copyrights of 10-15% of those files are owned by the GNU Project's Free Software Foundation.²³

IV. Analytic framework for operating system software copyright infringement.

The Copyright statute grants six exclusive rights, the first three of which are implicated in copying the Windows look and feel. The six exclusive rights are:

- (1) to reproduce the copyrighted work in copies or phonorecords;
- (2) to prepare derivative works based upon the copyrighted work;
- (3) to distribute copies or phonorecords of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending;
- (4) in the case of literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audiovisual works, to perform the copyrighted work publicly;
- (5) in the case of literary, musical, dramatic, and choreographic works, pantomimes, and pictorial, graphic, or sculptural works, including the individual images of a motion picture or other audiovisual work, to display the copyrighted work publicly; and
- (6) in the case of sound recordings, to perform the copyrighted work publicly by means of a digital audio transmission.²⁴

In order to determine whether a program in a Linux distribution infringed any of Microsoft's exclusive rights in its Windows OSs, Microsoft would have to prove:

“(1) ownership of a valid copyright, and (2) copying of constituent elements that are original.”²⁵

Microsoft's Windows operating system is a computer programs and programs are within the §101 definition²⁶ of “literary works” afforded copyright privilege by §102(a). The important inquiry however, is into the nature of the elements copied from Windows. Microsoft

²³ Red Hat 7.1 contained 1,016 packages which are programs or program libraries. Of those, 70 were programs written and/or maintained by the GNU project. Ming-Wei Wu and Ying-Dar Lin, Computer, Open Source Software Development: An Overview, Computer, a publication of the IEEE Computer Society. June 21, 2001. pp 32-38. Of the 42,585 free and open source software projects tracked by sourceforge.net, 30,241 use the GPL and 4,495 use the lesser GPL. See http://sourceforge.net/softwaremap/trove_list.php?form_cat=14

²⁴ 17 U.S.C. 106.

²⁵ *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 361 (1991). (Opinion by O'Connor. R. Ariz.)

²⁶ ““Literary works” are are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as... tapes, disks, or cards, in which they are embodied.” 17 U.S.C. 101.

would have to prove that the elements copied are privileged under the Copyright statute.

Microsoft would then have to show that the copyrighted elements were copied into an allegedly infringing program. This is the second prong of the Feist test is described in Lotus v. Borland. Microsoft would have to:

either present direct evidence of factual copying, or, if that is unavailable, evidence that the alleged infringer had access to the copyrighted work and that the offending and copyrighted works were so similar that the court may infer that there was factual copying (i.e., probative similarity). The plaintiff must then prove that the copying of the copyrighted material was so extensive that it rendered the offending works substantially similar.²⁷

With respect to XPde and other free software projects that set out to emulate the look and feel of the Windows OS, *proof* that something has been copied is almost beside the point. In order to encourage current users of Windows to switch to Linux, developers modify the way Linux looks and behaves so that a user who has become accustomed to operating a computer with Windows can operate a computer with Linux without having to relearn all the conventions and commands. The more relevant issue for Linux developers who want to offer Windows emulation is whether what the copying they want to do infringes Microsoft's copyright. Microsoft would have to prove more than the fact that something was copied from Windows. They would also have to prove copying of constituent elements of Windows that are original.²⁸ Not every constituent element of Windows is original or within the scope of copyright protection.

In analyzing computer software infringement, there is a confusing similarity in terminology. The four terms involved are “literal elements,” “non-literal elements” “literal copying” and “non-literal copying.” There are two things that can be copied; literal and non-literal elements of a program. Second, there are two ways in which these elements can be copied in two different ways: literal copying and non-literal copying. Literal elements are a program's literal expression whether in human readable source code or, if the source code has been compiled, in machine readable object code.²⁹ The words on a set of program menus are among a program's literal elements.³⁰ Non-literal elements include a programs “fundamental essence or structure.”³¹

These four terms yield three ways in which it is logically possible to copy a program so as to infringe copyright: literal copying of literal elements (almost the definition of infringement), non-literal copying of literal elements and non-literal copying of non-literal elements. The principal of non-contradiction makes a literal copy of a non-literal element an impossible option.

²⁷ Lotus, 49 F. 3d at 813 (internal citations omitted).

²⁸ Feist, 499 U.S. at 362.

²⁹ Computer Associates v. Altai, 982 F.2d 693, at 702. (2d Cir. 1992).

³⁰ Lotus, 49 F.3d at 814-815.

³¹ Computer Associates, 982 F.2d at 701.

A. Literal copying of literal elements.

Most software infringement cases involving literal elements involve copying of the most obvious literal elements of a program, its source and/or object code. That kind of copying is almost a non issue in the context of Windows to Linux graphical interface look and feel copying. Simply copying Windows object code would be pointless because it would not run in Linux.³² Windows is, for the most part written in the C++ computer language and Linux, again for the most part is written in C. Even if access to Windows source code were to be had, rewriting the code to make it work in Linux would be an order of magnitude more difficult than tweaking an existing Linux package.

The Copyright statute grants protection to “original works of authorship fixed in any tangible medium of expression.”³³ The bar for what constitutes originality under the statute is exceedingly low but not non-existent. If a program's interface includes novel features or shows creativity in the selection or arrangement of existing features, it has originality.³⁴ Distinctive icons and widgets³⁵ probably meet the bar for originality but the scope of protection is now pixel-thin. In Apple v. Microsoft, Apple sought broad copyright protection for the use of icons, program windowing, pull down menus and distinctive widgets. The Ninth Circuit rejected the broad claim saying that Apple could not use copyright to “get patent-like protection for the idea of a graphical user interface.”³⁶ In software “total concept and feel” cases, where the elements of a program are not individually protectable, only identical copying is barred.³⁷

Even identical copying of literal expression does not infringe copyright if what is being copied is not within the scope of the copyright privilege. This principle is encoded in §102 (b) of the Copyright Act:

In no case does copyright...extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.³⁸

³² Windows object code can be made to run under Linux using an open source implementation of the Windows API called Wine (www.winehq.com). Because the source code of projects like Wine is freely available and can be studied, it would be immediately transparent if illicit Windows code were being run under Linux.

³³ 17 U.S.C. 102(a).

³⁴ Melville B. Nimmer & David Nimmer, *Nimmer on Copyright*, §2.01[B]. (1997).

³⁵ A widget is:

“an element of a graphical user interface (GUI) that displays information or provides a specific way for a user to interact with the operating system and application. Widgets include icons, pull-down menus, buttons, selection boxes, progress indicators, on-off checkmarks, scroll bars, windows, window edges (that let you resize the window), toggle buttons, forms, and many other devices for displaying information and for inviting, accepting, and responding to user actions.”

IT Encyclopedia, <http://whatis.techtarget.com>

³⁶ Apple Computer, 35 F.3d at 1443.

³⁷ See Rachel v. Banana Republic, Inc., 831 F.2d 1503, 1507 (9th Cir. 1987). Case involving artificial African game animal heads, known as head mounts used by Banana Republic as part of safari theme decor.

Noninfringement found, applying the merger doctrine in holding that the idea of making an exact replica of an animal head merged with the expression of that idea in the head mount.

³⁸ 17 U.S.C. 102(b).

In Lotus, the First Circuit held that a set of program menus was the method by which the program was operated and therefore verbatim copies of 469 words in 50 menus did not infringe.³⁹ The court in Lotus held that where expression is part of a method of operation, even expression is not copyrightable.⁴⁰ Screen displays can be copyrighted separately as audiovisual works but insofar as they are also copyrightable literal elements of a program only some parts of the displays may be protectable.⁴¹

Taken together, the most likely elements that, if copied exactly, would constitute copyright infringement are distinctive or unique icons, images and words and sounds that are not part of a menu structure.⁴²

B. Non-literal copying of literal and non-literal elements.

The test for non-literal copying adopted by virtually every circuit is found in Computer Associates v. Altai. In that case, an Altai employee copied large parts of the source code from a Computer Associates program designed to interoperate with other operating systems. Once litigation brought the copying to the attention of Altai's management, the company had the borrowed code redacted and used a clean-room process to write new code that recreates the functionality. The original infringement was uncontested but Computer Associates also alleged that the new code still infringed its copyright in the structure and essence of the original program. The Second Circuit disagreed and used a three step analysis to significantly raise the bar in finding infringement of non-literal elements; abstraction, filtration and comparison.⁴³ The abstraction, filtration, comparison test evaluates the copyrightability of a program on multiple levels and then, once non-copyrightable material is filtered out, compares copyrightable elements against an allegedly infringing program.⁴⁴ The abstraction step is akin to reverse-engineering a program and

³⁹ Lotus, 49 F3d at 816.

⁴⁰ Id. The Supreme Court upheld Lotus in a per curiam decision that split 4-4 with one Justice not taking part. It may be noteworthy that at oral argument, Justice Ginsberg repeatedly returned to this point in criticizing the First Circuit's per se approach, characterizing it as an anomaly. The 4-4 split means that Lotus is not binding on the other circuits and the makeup of the court is unchanged so Lotus' tenuous stability remains.

⁴¹ Apple Computer, 35 F.3d at 1440-47.

⁴² The issues of trade mark and trade dress are beyond the scope of analysis for this version. They are, however, even greater sources of risk than copyright infringement for free software projects that copy the Windows look and feel too closely.

⁴³ Altai, 982 F.2d at 706-10.

⁴⁴ See Gates Rubber Co. v Bando Chem. Indus., Ltd., 9 F3d 823, 834-36 for a particularly thorough exposition of the process:

[A] computer program can often be parsed into at least six levels of generally declining abstraction: (i) the main purpose, (ii) the program structure or architecture, (iii) modules, (iv) algorithms and data structures, (v) source code, and (vi) object code.

The main purpose of a program is a description of the program's function or what it is intended to do. When defining a program's main purpose, the court must take care to describe the program's function as specifically as possible without reference to the technical aspects of the program.

The program's architecture or structure is a description of how the program operates in terms of its various functions, which are performed by discrete modules, and how each of these modules interact with each other. The architecture or structure of a program is often reduced to a flowchart, which a programmer uses visually to depict the inner workings of a program. Structure exists at nearly every level of a program and can be

requires the court to analyze a program on multiple levels.

In considering whether XPde and similar Linux desktops infringe copyrights in Windows, two things quickly become clear. First, applying the Altai test would be simple. Secondly, and for the same reason the test would be simple, it would almost inevitably lead to a finding that there is no non-literal copying by Linux that could infringe of the copyright in Windows.

If Microsoft were to complain about infringement, it would have to state which elements it believed to be infringing. Because of the heterogeneous nature of the respective code bases, at least three and probably four of the six levels of abstraction would be irrelevant. The two that would be irrelevant are object code and source code because of the different languages in which the programs are written. Even more telling than the different computer languages, the radically different ways that Windows and Linux handle graphics all but precludes copying the algorithms/data structures as well as modules from one to the other. Of the abstraction levels that remain, main function and program architecture, the former is a non-copyrightable idea and the latter inapposite.

Understanding why it would be all but impossible to find copyright infringement at the program architecture on Linux graphics requires a little understanding of the differences in the technology.

conceived of as including control flow, data flow, and substructure or nesting. Control flow is the sequence in which the modules perform their respective tasks. Data flow describes the movement of information through the program and the sequence with which it is operated on by the modules. Substructure or nesting describes the inner structure of a module whereby one module is subsumed within another and performs part of the second module's task.

The next level of abstraction consists of the modules. A module typically consists of two components: operations and data types. An operation identifies a particular result or set of actions that may be performed. For example, operations in a calculator program might include adding or printing data. A data type defines the type of item that an operator acts upon such as a student record or a daily balance.

Algorithms and data structures are more specific manifestations of operations and data types, respectively. An algorithm is a specific series of steps that accomplish a particular operation. Data structure is a precise representation or specification of a data type that consists of (i) basic data type groupings such as integers or characters, (ii) values, (iii) variables, (iv) arrays or groupings of the same data type, (v) records or groupings of different data types, and (vi) pointers or connections between records that set aside space to hold the record's values.

The computer program is written first in a programming language, such as Pascal or Fortran, and then in a binary language consisting of zeros and ones. Source code is the literal text of a program's instructions written in a particular programming language. Object code is the literal text of a computer program written in a binary language through which the computer directly receives its instructions.

These generalized levels of abstraction will not, of course, fit all computer codes. Ordinarily, expert testimony will be helpful to organize a particular program into various levels of abstraction. In any event, as pointed out earlier, the organization of a program into abstraction levels is not an end in itself, but it is only a tool that facilitates the critical next step of filtering out unprotectable elements of the program.

(Internal citations omitted).

Linux is unlike Microsoft Windows in that the operating system, graphical user interface, and window manager are separable. Microsoft locks these three programs together in Windows and makes it impossible to use a competing product for any one. If you use Windows at all, you use Microsoft's OS kernel, GUI and window manager. If you run the Linux kernel, you do not need to run a graphical interface at all and many users rely primarily or exclusively on a command line interface. When Linux-kernel based OSs use graphics at all, it is generally done with the X-windows system developed by MIT and originally used in Unix.

If the X program is run under Linux, it takes control of the screen from the command line and enables programs to create windows. X, requires another program, a window manager to actually manage the windows that X provides. Window managers in Linux generally give user the ability to customize the appearance of the interface. In Microsoft Windows, the ability to customize the interface is mostly limited to moving, resizing, or iconifying windows.

To restate the differences, the Linux (and UNIX) approach to providing a graphical interface is done with three separate components. In this case, there is the Linux OS itself, and running on top of that is the window system, X. Finally, there is a window manager that handles the decorative aspects of the interface. In contrast, the Microsoft Windows 2000 and XP OSs integrate the OS kernel with the graphical windowing system and window manager.

These systems can be made to look alike to an ordinary user but their underlying architectures are so different that they cannot be made to function alike nor can elements of one be easily grafted into the other. With code, very little is impossible but some things are so improbable as to be ridiculous. Making the source code, data structures, algorithms and modules of the Windows graphics system function in a Linux system fits into this category.

V. Conclusion

GNU/Linux based graphical interfaces that emulate the appearance of Microsoft's Windows operating system may, if they copy the exact appearance of unique or distinctive graphical elements infringe Microsoft's copyright. So long as they avoid exact copying of the relatively few unique or distinctive elements in the Windows interface, they can still provide an interface for Linux systems that Windows users are familiar with yet not infringe copyright. The cases that set out the relevant tests for determining whether they infringe are Lotus v. Borland and Apple v. Microsoft. Because of the differences between the two systems architectures, non-literal copying, as tested by Computer Associates v. Altai, is unlikely to be implicated. Trade mark and trade dress are areas of potentially serious legal risk for XPde and similar projects but is beyond the scope of this analysis.