LLVMLinux: x86 Kernel Build

Presented by:

Jan-Simon Möller

Presentation Date: 2012.08.30

Topics

- Common issues (x86 perspective)
- Specific Issues with Clang/LLVM
- Specific Issues with the Linux Kernel
- Status report on the x86 build
- Demo
- Q/A

Common issues

Common issues

- Recalling the issues in common:
  - Variable length arrays in structs (VLAIS)
    - A declaration like:

```
void f (int i) {
    struct foo_t {
        char a[i];
    } foo;
}
```

    - cannot be compiled in Clang, though declarations like:

```
void f (int i) {
    char foo[i];
}
```

    - are perfectly acceptable.

Common issues II

- Explicit register variables not supported

```
- register unsigned long
  current_stack_pointer asm("esp") __used;
+ #define current_stack_pointer (
  { unsigned long esp;
  asm("mov %%esp, %0" : "=r"(esp));
  esp; }
  )
```

Common issues III

- Segment references
  - More __refdata, __initdata, __exitdata attributes required
  - Investigate differences in linking and segments
  - Investigate module loading / unloading
- Linker / Kernel Linker Scripts experts – Ideas ?!

Issues With Building the Linux Kernel for X86 With Clang/LLVM

-

Clang

Unsupported in LLVM/Clang

- X86 specific flags support missing
- Big patch on integrated-as (macros)
- .code16gcc unsupported

Unsupported in LLVM/Clang

- **X86 specific opt-flags support missing**
  - -ffixed-REG
  - -fcall-used-REG
  - -fcall-saved-REG
- Workaround is CONFIG_ARCH_HWEIGHT_CFLAGS ***off*** (also patched out)

Unsupported in LLVM/Clang

- **Big patch on integrated-as (macros)**
  - On x86 the Integrated-Assembler (IA) is on by default.
  - Does not support all macro statements used in the kernel
- WIP upstream, still patches needed on Linux side. Currently we turn it off.

Unsupported in LLVM/Clang

- **.code16 / .code16gcc unsupported**
  - X86 bootup code needs real/16bit protected support
  - Backend missing in LLVM for X86-16

Issues With Building the Linux Kernel for X86 With Clang/LLVM

-

Kernel

Linux Kernel Patches

- Currently we apply a queue of 48 patches (split apart from a big patchfile) and most deal with:
  - Linking with integrated-as or disabling it
    - Necessary for some core & bootup code

arch/x86/boot/Makefile:

```
-    -Wall -Wstrict-prototypes \
-    -march=i386 -mregparm=3 \
+    -Wall -Wstrict-prototypes \
+    -Wno-unused-value -Wno-unused-parameter -mno-sse \
+    -march=i386 -mregparm=3 -no-integrated-as \
```

Linux Kernel Patches II

- Optimizations (e.g. currently -mno-sse)
- No support for 'register'

```
-register unsigned long current_stack_pointer
asm("esp") __used;
+#define current_stack_pointer (
{ unsigned long esp;
asm("mov %%esp, %0" : "=r"(esp)); esp; }
)
```

Linux Kernel Patches

- VLAIS

drivers/md/dm-crypt.c:

```
-   struct {
-         struct shash_desc desc;
-         char ctx[crypto_shash_descsize(lmk->hash_tfm)];
-   } sdesc;
+   char sdesc[sizeof(struct shash_desc) +
    crypto_shash_descsize(lmk->hash_tfm) +
    CRYPTO_MINALIGN] CRYPTO_MINALIGN_ATTR;
+   struct shash_desc *desc = (struct shash_desc *)sdesc;
```

Linux Kernel Patches

- Assembly: e.g. bt[c,r,s] → bt[c,r,s]l

```
-        asm volatile("bt %2,%1\n\t"
+        asm volatile("btl %2,%1\n\t"
```

Or

```
-        asm volatile("btc %2,%1\n\t"
+        asm volatile("btcl %2,%1\n\t"
```

Linux Kernel Patches

- Some more macro and alignment related patches
- Patch queue available in
  - [llvmlinux.git] / arch / all / patches /

    +
  - [llvmlinux.git] / arch / i586 / patches /

    or
  - [llvmlinux.git] / arch / x86_64 / patches /

Status of Building the Linux Kernel for x86 With Clang/LLVM

Status

- Prototype working build with tool from LLVMLinux repo
- Based on Kernel v3.3
- LLVM HEAD
- CLANG HEAD
- Booting to shell *and* desktop

Status II

- Driver modules work (even binary blobs) but unloading of driver modules fails
  - Might point to linking issues (e.g. __exit) with integrated-as
- .config is still limited

Status III

- Driver modules work (even binary blobs) but unloading of driver modules fails
  - Might point to linking issues (e.g. __exit) with integrated-as
- .config is still limited

Demo

- Demo on x86_64 system
  - Built with LLVMLinux tool
  - I7-2 CPU
  - Desktop system
  - Nvidia GPU

What's Left to Do?

Todos

- Finish work on macro support
- Split patches and rebase / apply to Linux/LLVM/Clang HEAD
- code16(gcc) support for LLVM/Clang
- Segment linkage differences
  - Cross-check linkage with module (un)loading

Goals / Plans

- Get an all-upstream-HEAD build working
- Add it to buildbot
- Support forward-port / split / rebase efforts
- Get code merged / issues solved upstream
- Extend coverage and enable full distro .config
- Easy compile wrappers

Bugs open / Roadmap

- Tracker bug in http://llvm.org/bugs
  - Bug 4068 - [META] Compiling the Linux kernel with clang
- http://llvm.linuxfoundation.org/index.php/Roadmap

How Can I Help?

- Review patches for Clang/LLVM and Kernel
- Help get all patches split/rebased/upstream
- Try the code on your own HW
- Propose new test cases
- Report Bugs
- Work on unsupported features and Bugs
- New targets and Arch support

Who wouldn't want a penguin with wings?

Questions ?

Thank you !

http://llvm.linuxfoundation.org

Contribute to the LLVMLinux Project

- Project wiki page
  - http://llvm.linuxfoundation.org
- Project Mailing List
  - http://lists.linuxfoundation.org/mailman/listinfo/llvmlinux
  - http://lists.linuxfoundation.org/pipermail/llvmlinux/
- IRC Channel
  - #llvmlinux on OFTC
  - http://buildbot.llvm.linuxfoundation.org/irclogs/OFTC/%23llvmlinux/

LLVMLinux: x86 Kernel Build

Presented by:

Jan-Simon Möller

Presentation Date: 2012.08.30

Topics

- Common issues (x86 perspective)
- Specific Issues with Clang/LLVM
- Specific Issues with the Linux Kernel
- Status report on the x86 build
- Demo
- Q/A

Common issues

Common issues

- Recalling the issues in common:
  - Variable length arrays in structs (VLAIS)
    - A declaration like:

```
void f (int i) {
    struct foo_t {
        char a[i];
    } foo;
}
```

    - cannot be compiled in Clang, though declarations like:

```
void f (int i) {
    char foo[i];
}
```

    - are perfectly acceptable.

Common issues II

- Explicit register variables not supported

```
- register unsigned long
  current_stack_pointer asm("esp") __used;
+ #define current_stack_pointer (
  { unsigned long esp;
  asm("mov %%esp, %0" : "=r"(esp));
  esp; }
  )
```

Common issues III

- Segment references
  - More __refdata, __initdata, __exitdata attributes required
  - Investigate differences in linking and segments
  - Investigate module loading / unloading
- Linker / Kernel Linker Scripts experts – Ideas ?!

A lot of segment warnings (silenced by __refdata and others) – probably not the final solution. We need to investigate.

Issues With Building the Linux Kernel for X86 With Clang/LLVM

-

Clang

Unsupported in LLVM/Clang

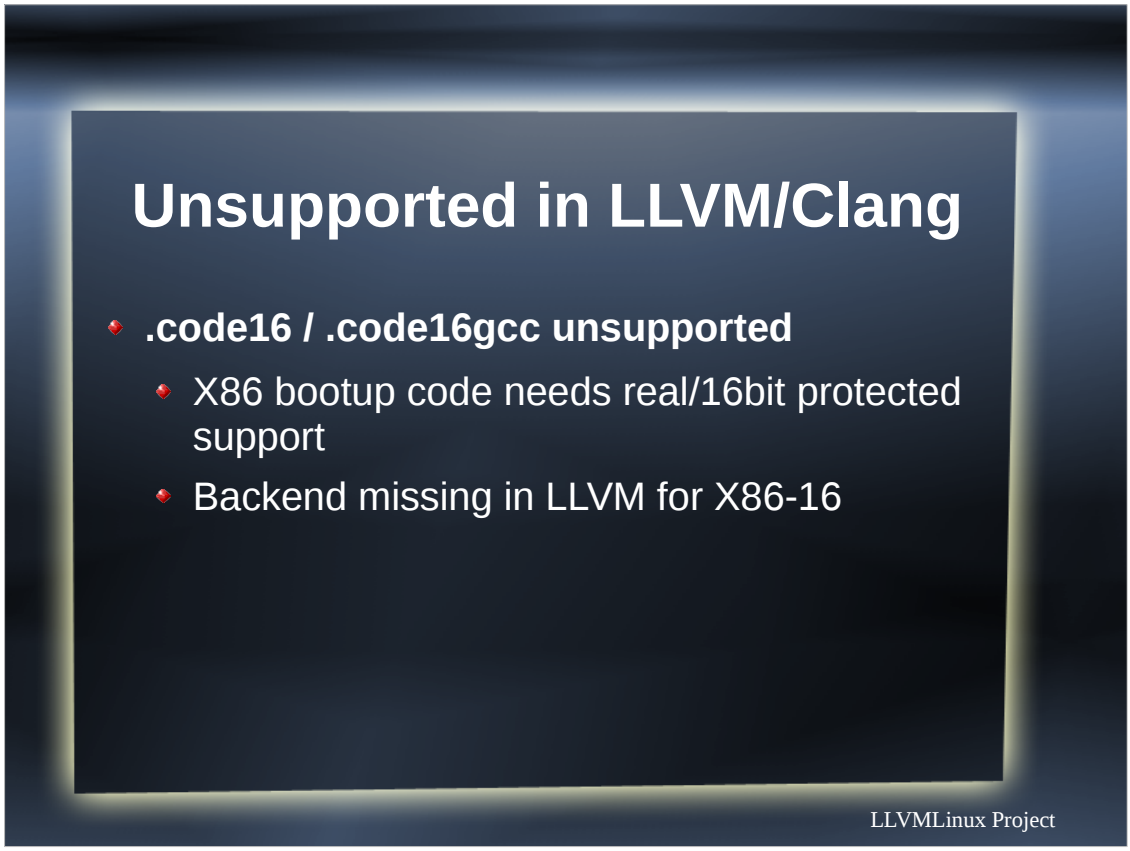
- X86 specific flags support missing
- Big patch on integrated-as (macros)
- .code16gcc unsupported

Unsupported in LLVM/Clang

- **X86 specific opt-flags support missing**
  - -ffixed-REG
  - -fcall-used-REG
  - -fcall-saved-REG
- Workaround is CONFIG_ARCH_HWEIGHT_CFLAGS ***off*** (also patched out)

Unsupported in LLVM/Clang

- **Big patch on integrated-as (macros)**
  - On x86 the Integrated-Assembler (IA) is on by default.
  - Does not support all macro statements used in the kernel
- WIP upstream, still patches needed on Linux side. Currently we turn it off.



There is no code16(gcc) support in upstream clang – no prio from upstream.

Issues With Building the Linux Kernel for X86 With Clang/LLVM - Kernel

LLVMLinux Project

Linux Kernel Patches

- Currently we apply a queue of 48 patches (split apart from a big patchfile) and most deal with:
  - Linking with integrated-as or disabling it
    - Necessary for some core & bootup code

arch/x86/boot/Makefile:

```
-    -Wall -Wstrict-prototypes \
-    -march=i386 -mregparm=3 \
+    -Wall -Wstrict-prototypes \
+    -Wno-unused-value -Wno-unused-parameter -mno-sse \
+    -march=i386 -mregparm=3 -no-integrated-as \
```

Linux Kernel Patches II

- Optimizations (e.g. currently -mno-sse)
- No support for 'register'

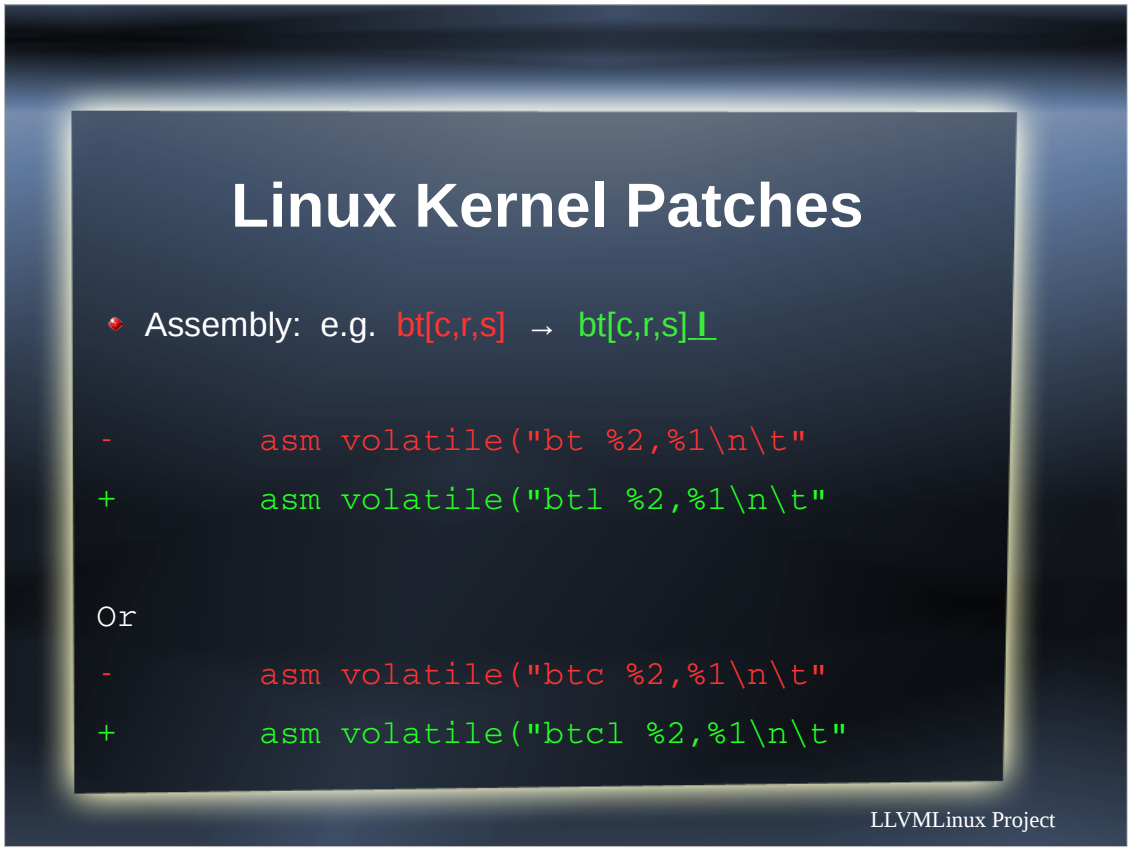
```
-register unsigned long current_stack_pointer
asm("esp") __used;
+#define current_stack_pointer (
{ unsigned long esp;
asm("mov %%esp, %0" : "=r"(esp)); esp; }
)
```

Linux Kernel Patches

- VLAIS

drivers/md/dm-crypt.c:

```
-   struct {
-         struct shash_desc desc;
-         char ctx[crypto_shash_descsize(lmk->hash_tfm)];
-   } sdesc;
+   char sdesc[sizeof(struct shash_desc) +
    crypto_shash_descsize(lmk->hash_tfm) +
    CRYPTO_MINALIGN] CRYPTO_MINALIGN_ATTR;
+   struct shash_desc *desc = (struct shash_desc *)sdesc;
```



http://llvm.org/bugs/show_bug.cgi?id=9362

Eli Friedman 2011-03-02 14:53:11 CST

(In reply to comment #5)
> If the bit base operand specifies a memory location, the operand represents
> the address of the byte in memory that contains the bit base (bit 0 of the
> specified byte) of the bit string. The range of the bit position that can be
> referenced by the offset operand depends on the operand size.

Precisely; and since the behavior varies depending on the operand size, and we
can't deduce the operand size, it's an error. (Note that by the time we
actually parse the asm, it's impossible to tell anything about the size of the
memory operand.)

Linux Kernel Patches

- Some more macro and alignment related patches
- Patch queue available in
  - [llvmlinux.git] / arch / all / patches /

    +
  - [llvmlinux.git] / arch / i586 / patches /

    or
  - [llvmlinux.git] / arch / x86_64 / patches /

Status of Building the Linux Kernel for x86 With Clang/LLVM

LLVMLinux Project

Status

- Prototype working build with tool from LLVMLinux repo
- Based on Kernel v3.3
- LLVM HEAD
- CLANG HEAD
- Booting to shell *and* desktop

Status II

- Driver modules work (even binary blobs) but unloading of driver modules fails
  - Might point to linking issues (e.g. __exit) with integrated-as
- .config is still limited

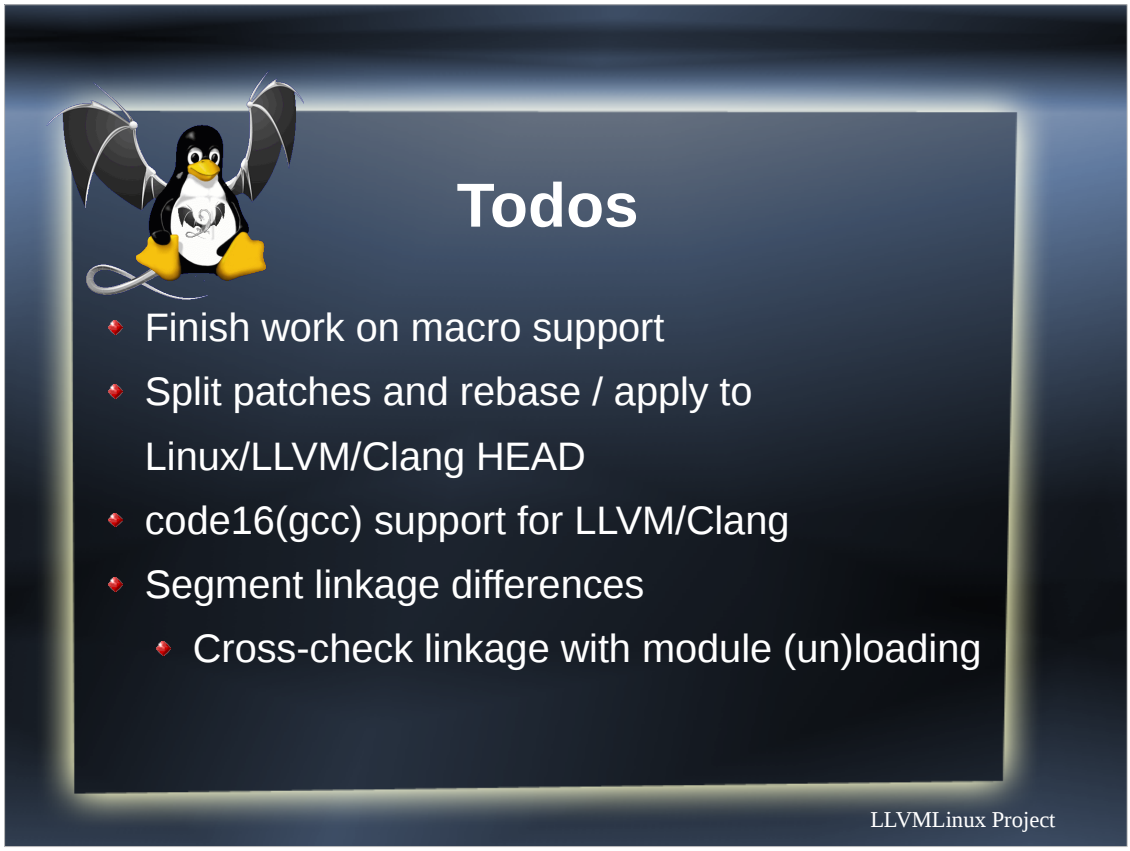
Status III

- Driver modules work (even binary blobs) but unloading of driver modules fails
  - Might point to linking issues (e.g. __exit) with integrated-as
- .config is still limited

Demo

- Demo on x86_64 system
  - Built with LLVMLinux tool
  - I7-2 CPU
  - Desktop system
  - Nvidia GPU

What's Left to Do?



Not a definitive list.

There is a lot more on the Roadmap page on the wiki.

Not a definitive list.

There is a lot more on the Roadmap page on the wiki.

Bugs open / Roadmap

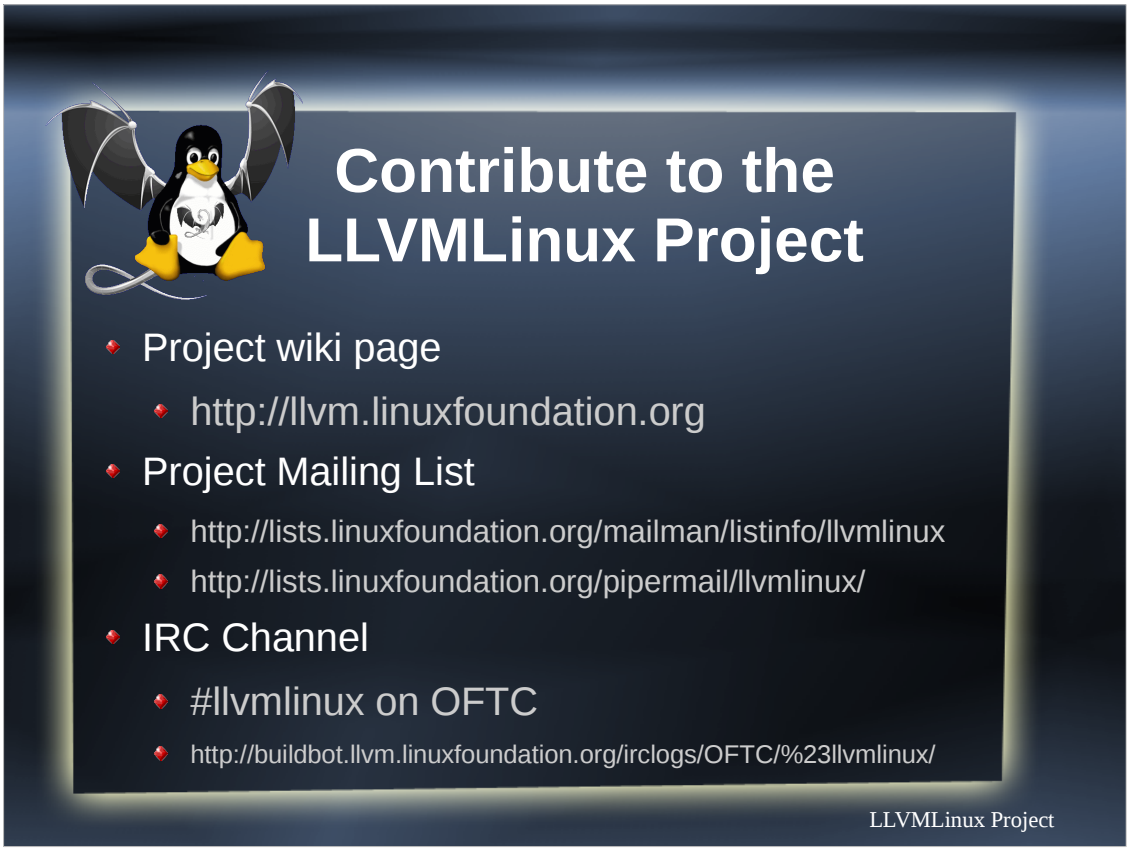
- Tracker bug in http://llvm.org/bugs
  - Bug 4068 - [META] Compiling the Linux kernel with clang
- http://llvm.linuxfoundation.org/index.php/Roadmap

Last real slide.

We can take our time here.

End of slide deck.

Recap contact slide is next.



Leave this up at the end.