The unified x86 tree and clang

Issues and Solutions compiling ./arch/x86/ with clang

Presenter:

Jan-Simon Möller

jansimon.moeller@gmx.de

IRC: #llvmlinux @ OFTC

llvm.linuxfoundation.org

Current status

- LLVM/Clang 3.3 works w/o any patches
- The Kernel still needs our patchset
- We can build and run on x86 with nearly most options enabled.
  - See “Broken_kernel_options” on http://llvm.linuxfoundation.org/

Current status

- x86_64 builds and boots
- Try:

```
git clone \

  http://git.linuxfoundation.org/llvmlinux.git

cd llvmlinux/targets/x86_64

make

(or to save time: make CLANG_TOOLCHAIN=prebuilt)
```

~~The Problems~~ (solved)

- Assembly: BT* -> BT*L e.g. BTC vs. BTCL
  - was not allowed in clang (no defaults or ‘guessing’ of length)
  - solved in llvm/clang since r186904
  - llvm bug # 9365

~~The Problems~~ (solved)

- arch/x86/include/asm/uaccess.h

```
register __inttype(*(ptr)) __val_gu asm("%edx");
```

  - gcc uses %edx as starting point also for 64bit using the registers as pair
  - clang does check the size and bails out
  - solved in the Kernel using the macro _ASM_DX (expands to edx on 32 and rdx on 64)
  - Problem was actually solved in the kernel already for a different use-case!

The Problems

- We disabled clang's integrated assembler (and rely on gas for linking)
  - Reasons:
    - no .code16 support
    - inline assembly handling

```
arch/x86/boot/Makefile
+# For clang we need to rely on no-integrated-as for .code16
+ifeq ($(COMPILER),clang)
+KBUILD_CFLAGS  += -Wno-unused-value -Wno-unused-parameter \
+                -mno-sse $(call cc-option,-no-integrated-as,)
+endif
```

The Problems

- We disabled clang's integrated assembler (and rely on gas for linking)
  - Reasons:
    - no .code16 support
    - inline assembly handling

arch/x86/realmode/rm/Makefile

```
+# For clang, we need to enforce not to use the integrated assembler
+AFLAGS_wakeup_asm.o   = $(call cc-option,-no-integrated-as,)
+AFLAGS_bioscall.o     = $(call cc-option,-no-integrated-as,)
+AFLAGS_copy.o         = $(call cc-option,-no-integrated-as,)
+AFLAGS_trampoline_32.o = $(call cc-option,-no-integrated-as,)
+AFLAGS_trampoline_64.o = $(call cc-option,-no-integrated-as,)
+AFLAGS_reboot_32.o    = $(call cc-option,-no-integrated-as,)
```

The Problems

- Odd use of __builtin_constant_p
  - Wait/wound mutexes (see http://lwn.net/Articles/548909/)

```
kernel/mutex.c:
@@ -450,7 +450,7 @@
-   if (!__builtin_constant_p(ww_ctx == NULL) &&
                  ww_ctx->acquired > 0) {
+   if (!(ww_ctx == NULL) &&
                   ww_ctx->acquired > 0) {
```

According to author it is an optimization (with gcc).
Clang does not support this use of __builtin_constant_p .

The Problems

- Clang does not support named registers. This affects:

```
arch/x86/include/asm/thread_info.h

 /* how to get the current stack pointer from C */
-register unsigned long current_stack_pointer asm("esp") __used;
+#define current_stack_pointer ({ \
+       unsigned long esp; \
+       asm("mov %%esp, %0" : "=r"(esp)); \
+       esp; \
+})
```

Problem statement

- Mid-term, we will re-enable clangs integrated assembler. This will need some more changes to work out.
  - ASM syntax issues
  - 32bit/64bit size (like %eax -> %_ASM_DX)

- >75% of the Kernel compiles/links with integrated-as

Problem statement

- How do we approach these issues?
  - There are concerns regarding maintainability (ifdef's, extra code, …)
  - The ideal solution works for both compilers
  - We need to find the proper place

End

Thanks.

http://llvm.linuxfoundation.org

#llvmlinux on OFTC