



LLVMLinux project

LLVMLinux: Patch status

Presented by:
Behan Webster
(LLVMLinux project lead)

Presentation Date: 2014.10.17



The LLVM Linux Project Goals

- Fully build the Linux kernel for multiple architectures, using the Clang/LLVM toolchain
- Discover LLVM/Kernel issues early and find fixes quickly across both communities
- Upstream patches to the Linux Kernel and LLVM projects
- Bring together like-minded developers
- Enable the kernel community to do more in depth analysis of the kernel code



Patched Mainline Kernel Tree

- A mainline kernel tree with all LLVM Linux patches applied on top is now available:
 - `git://git.linuxfoundation.org/llvmlinux/kernel.git`
- Dated llvmlinux branches
 - `remotes/origin/llvmlinux-2014.10.15`
- The master branch is rebased regularly
- Also a part of `linux-next` and 0-day kbuild-robot



LLVMLinux Project Status

- LLVM/clang:
 - All LLVMLinux patches for LLVM are Upstream
 - Newer LLVM patches to support the Linux kernel are mostly being added by upstream maintainers
 - Named register support for sp in clang-3.6-svn
 - A number of ASM related issues



LLVMLinux Kernel Patches

Architecture	Number of patches	Submitted	Acked	Accepted Upstream
all	27	8	3	2
arm	17	13	9	3
aarch64	3	0	0	0
x86_64	2	0	0	0
TOTALS	49	21	12	5



Integrated Assembly Status

- David Woodhouse added .code16 support for X86 ASM
- Renato Golin, Vicinius Tinti, Saleem Abdulrasool and Stepan Dyatkovskiy are working on fixing IA issues in clang to support the Linux ARM kernel code (and ultimately AARCH64)
- For now we disable the IA and use GNU as instead



Different option passing

- gcc passes -march to GNU as
- clang doesn't... (Bug submitted PR)

-CFLAGS_aes-ce-cipher.o += -march=armv8-a+crypto

+CFLAGS_aes-ce-cipher.o += -march=armv8-a+crypto -Wa,-march=armv8-a+crypto



extern inline: Different for gnu89 and gnu99

- GNU89/GNU90 (used by gcc)
 - Function will be inlined where it is used
 - No function definition is emitted
 - A non-inlined function may also be provided
- GNU99/C99 (used by clang)
 - Function will be inlined where it is used
 - An external function is emitted
 - No other function of the same name may be provided.
- Solution? Use “static inline” instead.
- Only still an issue for ARM support for ftrace (submitted)



Attribute Order

- gcc is less picky about placement of `__attribute__(())`
- clang requires it at the end of the type or variable

```
-struct __read_mostly va_alignment va_align = {  
+struct va_alignment __read_mostly va_align = {
```



ARM percpu patch

- One of the uses of Named Registers in the ARM code is due to a deficiency in gcc.
- The new code which works with gcc fails in clang.
- Solution, provide routines for both, and choose at compile time
- Gcc:

```
asm("mrc p15, 0, %0, c13, c0, 4" : "=r" (off) : "Q" (*sp));
```

- Clang:

```
asm("mrc p15, 0, %0, c13, c0, 4" : "=r" (off) :: "memory");
```



Section Mismatch Issues (MergedGlobals)

- By default clang merges globals with internal linkage into one: MergedGlobals
- Allows globals to be addressed using offsets from a base pointer
- Can reduce the number of registers used
- Modpost uses symbol names to look for section mismatches
- MergedGlobals breaks modpost (false positive section mismatches)
- Current solution: use -mno-global-merge to stop global merging
- Updates to modpost may allow this optimization to be enabled again



ARM eabi support

- Clang emits code which uses the “aeabi” ARM calls which are implemented in compiler-rt (equivalent to libgcc)
- Compiler-rt doesn't easily cross compile yet...

```
void __aeabi_memcpy(void *dest, const void *src, size_t n)
```

```
void __aeabi_memmove(void *dest, const void *src, size_t n)
```

```
void __aeabi_memset(void *s, size_t n, int c)
```



Variable Length Arrays In Structs

- VLAIS isn't supported by Clang (undocumented gcc extension)

```
char vla[n];                                /* Supported, C99/C11 */

struct {
    char flexible_member[ ]; /* Supported, C99/C11 */
} struct_with_flexible_member;

struct {
    char vla_is[n];      /* Explicitly not allowed by C99/C11 */
} variable_length_array_in_struct;
```

- VLAIS is used in the Linux kernel in a number of places, spreading mostly through reusing patterns from data structures found in crypto



VLAIS Removal Example

```
- struct {
-     struct shash_desc shash;
-     char ctx[crypto_shash_descsize(tfm)];
- } desc;
+ char desc[sizeof(struct shash_desc)
+             + crypto_shash_descsize(tfm)] CRYPTO_MINALIGN_ATTR;
+ struct shash_desc *shash = (struct shash_desc *)desc;
unsigned int i;

- desc.shash.tfm = tfm;
+ shash->tfm = tfm;
```

(from crypto/hmac.c)

LLVMLinux project



VLAIS Removal Example

```
- struct {  
-     struct shash_desc shash;  
-     char ctx[crypto_shash_descsize(tfm)];  
- } desc;  
+ SHASH_DESC_ON_STACK(shash, tfm);  
  
- desc.shash.tfm = tfm;  
+ shash->tfm = tfm;
```



VLAIS Removal Example (cont.) (the missing pieces)

```
#define ARCH_KMALLOC_MINALIGN __alignof__(unsigned long long)

#define CRYPTO_MINALIGN ARCH_KMALLOC_MINALIGN

#define CRYPTO_MINALIGN_ATTR __attribute__((__aligned__(CRYPTO_MINALIGN)))

struct shash_desc {
    struct crypto_shash *tfm;
    u32 flags;
    void *__ctx[] CRYPTO_MINALIGN_ATTR;
};
```



Bloat-o-meter gcc vs clang

	text	data	bss	dec	hex	filename
5451700	230636	153632	5835968	590cc0	build/targets/vexpress/build/kernel-gcc/vmlinux	
6178530	229100	154058	6561688	641f98	build/targets/vexpress/build/kernel-clang/vmlinux	

Bloat-o-meter:

add/remove: 1781/996 grow/shrink: 8172/4922 up/down: 585929/-413928 (172001)

function	old	new	delta
<u>blockdev_direct_IO</u>	5364	15052	+9688
<u>cache_slow_downcall.write_buf</u>	-	8192	+8192
<u>do_con_write</u>	-	7572	+7572
...			
<u>do_write_buffer</u>	8008	-	-8008
<u>write_buf</u>	8192	-	-8192
<u>__func__</u>	11219	-	-11219



Contribute to the LLVM Linux 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/>
- LLVM Linux Community on Google Plus