

Debug and develop
uClibc
using QEMU

Khem Raj

Embedded Linux Conference

11-13 April 2011, San Francisco

Agenda

- QEMU Setup
- Compiling uClibc for debugging
- Debugging uclibc dynamic linker using QEMU
- Some more on gdb
- Q & A

What is QEMU ?

- Processor emulator
 - Emulates ARM, x86, powerpc, mips, SH ...
 - Has a built-in GDB stub
- Getting QEMU
 - <http://bellard.org/qemu/>
 - Your favourite distribution might have already built it for you

Enable GDB stub

- QEMU options
 - -s enables the gdb stub
 - -S instructs QEMU to stop after system restart
 - Waits for gdb to connect
 - -gdb tcp::1234
 - Enables port 1234 on host

Debugging uClibc

- Use printf debugging
 - Compile with LD_DEBUG_EARLY
- Use JTAG with debugger e.g. BDI
 - Expensive
- Use emulators
 - QEMU

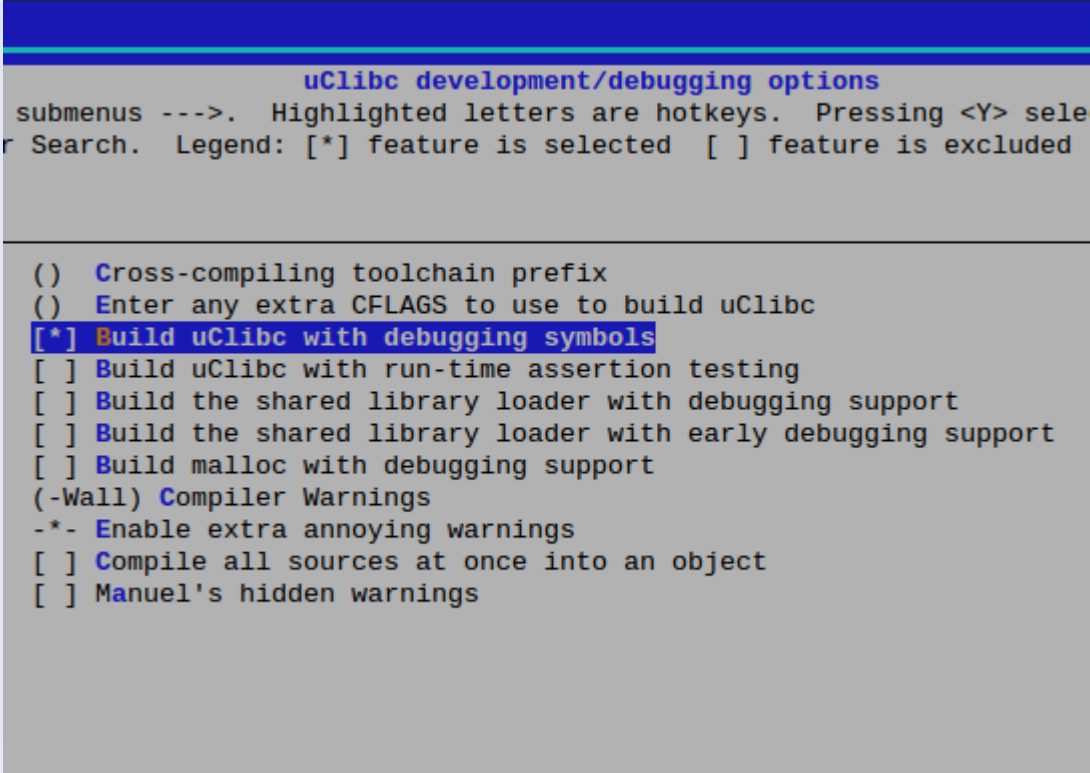
Debugging ld.so

- Dynamic linker is one of first userspace program started after kernel boots.
- Dynamic linkers rely on printf debugging
- Helpful in porting to new architectures
- Debugging functionality
- fixing bugs

Compiling uClibc for debugging

- Compile uclibc with debug information

```
make CROSS=/scratch/oe/gemuarmlinux-uclibceabi- menuconfig
```



```
uClibc development/debugging options
submenus --->. Highlighted letters are hotkeys. Pressing <Y> sele
r Search. Legend: [*] feature is selected [ ] feature is excluded

() Cross-compiling toolchain prefix
() Enter any extra CFLAGS to use to build uClibc
[*] Build uClibc with debugging symbols
[ ] Build uClibc with run-time assertion testing
[ ] Build the shared library loader with debugging support
[ ] Build the shared library loader with early debugging support
[ ] Build malloc with debugging support
(-Wall) Compiler Warnings
-*- Enable extra annoying warnings
[ ] Compile all sources at once into an object
[ ] Manuel's hidden warnings
```

Debugging ld.so

- Find out start address offset of ld.so

```
objdump -f ld-uClibc.so.0 |grep start
```

```
start address 0x00000ed0
```

- Find the virtual address mapping of ld.so
 - gdb's command `info shared`
 - Use `SUPPORT_LD_DEBUG_EARLY` which dumps the address
 - Gdb command `info proc mapping` or reading `proc/<pid>/maps`
- Add start address with virtual address to get the final address to load symbol information

Setup debugging environment

- Launch QEMU system emulation

```
qemu-system-arm -M versatilepb -m 256 -gdb tcp::1234 -s -S  
-kernel <kernel> -drive file=<image> -append  
'console=ttyAMA0 console=ttyS0 root=/dev/sda rw debug  
user_debug=-1'
```

- Launch cross gdb in another terminal

```
arm-oe-linux-uclibceabi-gdb
```

- Connect to waiting QEMU

```
(gdb) target remote :1234  
Remote debuggin using :1234  
0xc001eb30 in caliberate_delay()
```

Debugging ld.so

- Launch QEMU system emulation

```
qemu-system-arm -M versatilepb -m 256 -gdb tcp::1234 -s -S  
-kernel <kernel> -drive file=<image> -append  
'console=ttyAMA0 console=ttyS0 root=/dev/sda rw debug  
user_debug=-1'
```

- Use `add-symbol-file <address>` to load the debug info to right address.
- Set breakpoint in `__dl_get_ready_to_run ()`

```
(gdb) b __dl_get_ready_to_run
```

```
Breakpoint 1 at 0x40005f94: file ldso/ldso/ldso.c, line  
366.
```

Debugging ld.so

- Connect to remote target
- 'Continue' should hit the breakpoint in ld.so

```
Breakpoint 1, _dl_get_ready_to_run (tpnt=0x400a9730, load_addr=0x40007158, auxvt=0x0, envp=0x4000f030, argv=0x40006b24) at ldso/ldso/ldso.c:366
366         nextp = unsecure_envvars;
(gdb) c
Continuing.
```

```
Breakpoint 1, _dl_get_ready_to_run (tpnt=0x400a9730, load_addr=0x40007158, auxvt=0x0, envp=0x4000f030, argv=0x40006b24) at ldso/ldso/ldso.c:366
366         nextp = unsecure_envvars;
(gdb)
Continuing.
```

.gdbinit

- Convenience

```
File Edit View Scrollback Bookmarks Settings Help
get output-radix 16
add-symbol-file /scratch/oe/qemuarm/work/qemuarm-oe-linux-uclibceabi/uclibc-0.9.31+gitrf26c5f6952ce9bf8edec9c1571c47addb1bcc442-r34.0/git/lib/ld-uClibc.so.0 0x40000000+0xed0
add-symbol-file /scratch/oe/build/work/qemumips-oe-linux-uclibc/sysvinit-2.86-r57/sysvinit-2.86/src/init 0x401570

define loadlibc
    add-symbol-file /scratch/oe/qemuarm/work/qemuarm-oe-linux-uclibceabi/uclibc-0.9.31+gitrf26c5f6952ce9bf8edec9c1571c47addb1bcc442-r34.0/git/lib/ld-uClibc.so.0 $arg0+0xed0
end

#b *0x2aba24d4
#b *(0x2aafb000 + 0x5aad8)
#b _dl_perform_mips_global_got_relocations
#b _dl_parse_relocation_information
#b elfinterp.c:217
#b ldso.c:1003
#b __dl_runtime_resolve
#b vsyslog
#b __uClibc_main
#b _dl_get_ready_to_run
#b __start
target remote localhost:1234
c
c
c
c
```

Frontends to gdb

- Data Display Debugger (DDD)
 - Uses gdb to control the target
 - Provided rich GUI experience
- Eclipse CDT
- Kdevelop
- Insight

Q & A

Happy Debugging