Software Debugging

Albert Ou

UC Berkeley

<aou@eecs.berkeley.edu>

Debugging with Spike

- Advantages:
  - Determinism
  - Reliably correct execution
  - High visibility of architectural state
  - Simplicity of obtaining instruction traces
  - Single-stepping

- Disadvantages
  - Determinism
    - Hides concurrency bugs
  - No knowledge of higher-level software abstractions
    - Preemptive multitasking
    - Multiple virtual address spaces
  - Unsuitable for directly debugging user-level processes

Spike: Interactive Mode

- Invoked with -d flag or SIGINT (^C)
- reg [core_id] <register>
  - print x-register value, where register is either an ABI name (e.g., ra, s0) or a numeric index
- fregs [core_id] <register>
  - print f-register as single-precision value
- dregs [core_id] <register>
  - print f-register as double-precision value

Spike: Interactive Mode

- mem <addr>
  - print value at address; if core_id is omitted, treat as physical address
- str <addr>
  - print NUL-terminated string at physical address
- until reg|mem|pc <val>
  - run silently until reg/mem/pc equals the given value
- r
  - run/resume execution verbosely
- rs
  - run silently

Case Study: Porting the Linux Kernel

- Initial port is arguably most difficult: first major exercise of gcc and glibc
  - Many adventures to reminisce about – come see me for details

- Kernel mapped into the top of every virtual address space above PAGE_OFFSET

Kernel Hacking

- CONFIG_EARLY_PRINTK
  - Bare-bones serial console driver
  - Primary method of retrieving dmesg(8) output before TTY subsystem is fully initialized

- CONFIG_FRAME_POINTER
  - dump_stack()
  - “Naked” kernel-mode stack backtracing simplified by
    - Consistent use of s0 as the frame pointer
    - Fixed location of sp on the stack frame
    - Absence of branch delay slots
  - Avoids heuristics
  - Current limitation: cannot continue backtrace across exceptions; requires interpretation of pt_regs structure

Kernel Hacking

- CONFIG_DEBUG_INFO
  - DWARF4: open standard format for source-level debugging; only slightly complicated by linker relaxations

- Convert PC into file name and line number:

```
addr2line -e vmlinux -fp <addr>
```

- Disassembly with source interspersed:

```
objdump -dS vmlinux
```

Debugging with the Proxy Kernel

- Intended for testing self-contained kernels
  - Enables tractable waveform dumps in situations where OS boot overhead is prohibitive (e.g., RTL emulation)
  - Major feature: printf()
- Dependence on minimal infrastructure
- Supports dynamic linking
  - Simpler environment to analyze ld.so

GNU Debugger

- Original RISC-V port contributed by Todd Snyder (Bluespec, Inc.)
- Recent work at UCB:
  - Tracking upstream trunk of unified binutils-gdb repository
  - Updated to the most recent ABI
  - Added core debugging target and Linux native support
- Preferred *in situ* debugging method once kernel and dynamic linker are reasonably stable

GDB: Core Target

- Linux kernel
  - Emitting ELF core dumps involves some architecture-dependent handlers
  - Exports register sets in .notes section
    - Canonical NT_PRSTATUS note: “general-purpose” registers
    - NT_PRFREG note: floating-point registers
    - Can define architecture-specific note types and register views for extended state
  - Repurposes mechanisms used for PTRACE_{GET,SET}REGS
- BFD (binutils)
  - Converts notes into “.reg” pseudo-sections
  - elf_backend_grok_prstatus(), elf_backend_grok_psinfo()
- GDB
  - Interprets opaque data and populates inferior