



Learning gem5 – Part N

Other things not covered here

Jason Lowe-Power

<http://learning.gem5.org/>

<https://faculty.engineering.ucdavis.edu/lowepower/>



Full system support

Full system is like a virtual machine.

gem5 exposes a “bare metal” interface

Requires a kernel, disk image, etc.

See <http://learning.gem5.org/book/part5/> for simple x86 example

Full system vs Syscall emulation

Full system (FS) mode:

- Runs unmodified OS
- Like QEMU/hypervisor
- Emulates or models all devices
- Some ISAs supported better than others

Syscall emulation (SE) mode:

- Runs user-mode binaries
- Decoder catches syscalls
 - pseudo instruction
- Emulates the effect
 - sim/syscall_emul
- Different impl. for each ISA

GPU and device models

AMD recently released a HSAIL GPU model (src/gpu-compute)

Many devices supported for FS simulation

- Ethernet (and multi-system simulation)

- VNC for graphics

- IDE controllers for disks

- No Mali GPU for ARM

- VirtIO

Most devices are functional-only

Other features

Probes and tracing

Remote GDB

Dynamically-linked binaries in SE mode

Power modeling and PMU

And many, many, many others

Caveats

gem5 is a tool, not a panacea

Most models are not validated against “real” hardware

See “Architectural Simulators Considered Harmful”

<https://doi.org/10.1109/MM.2015.74>

There are bugs!



Getting (more) help

Main gem5 wiki: <http://gem5.org/>

My book:

<http://learning.gem5.org>

https://github.com/powerjg/learning_gem5

Mailing lists: http://gem5.org/Mailing_Lists

gem5-users: General user questions
(you probably want this one)

gem5-dev: Mostly code reviews and high-level
dev talk

