

Minutes from sponsor meeting held at Intel on 5/2/12:

ARs:

- **General project requirement:** determine how to grab data structures inside the driver and dump the information safely.
- **General project requirement:** determine how interrupts are configured on the network device.
- **General project requirement:** determine which network device is modeled in QEMU, and go out to the web and find data sheet for this device.
- **General project requirement:** understand debugfs.
- **General project requirement:** setup basic development framework for debugfs.
- **General project requirement:** setup basic E1000 device.
- **General project requirement:** compile E1000.
- **Everyone:** set up a working QEMU virtual machine for development use.
- **Kyle:** email script for QEMU virtual machine setup.
- **David:** follow Kyle's virtual machine setup script and compile a copy of the custom kernel we will be using.
- **Everyone:** look at the E1000 driver code (specifically, look for the 'adapter' data structure in the driver code).

- Meeting started at 5:00 p.m.
- Shannon: It might not matter which virtual machine system we use (VirtualBox, QEMU, etc.)
- PJ: QEMU has certain features that are not available in other systems. We should be using QEMU for our development.
- Shannon listed a few general aspects of setting up a virtual machine.
- We will be developing our project based around the E1000 linux driver.
- We want to build a framework that can be used for almost any network device.
- Interactions with the debugfs system should be in a separate file in the driver source.
- debugfs requires special header files.
- We need to determine how to grab data structures inside the driver and dump the information safely.
- adapter struct, hw struct, and other specific structures are used by the E1000 driver.
- Facilities exist in the E1000 driver that already provide safe data access.
- We need to determine how interrupts are configured on the device.
- We should determine what device is modeled in QEMU, and go out to the web and find data sheet for this device.
- QEMU makes the development process much faster.
- We can simulate driver and hardware errors by screwing up the simulated device registers.
- The E1000 driver code uses a large array of register structures.
- Adapter struct - common for E1000
 - Collect internal data into adapter struct

- Address space, how many errors, how many bits set/received, ...
 - Global data structure
- E1000 layout: OS generic (top) -> network core -> adapter struct
- First steps for our project - understand debugfs, setup basic framework for debugfs, setup basic E1000 device, compile E1000.
- Concentrate on the E1000 driver version in the kernel tree.
- Our project result should work for the kernel version of the driver.
- Project validation? - Hard to test hardware for E1000 1Gbps
- Sponsors will look at our code and give us feedback.
- We should test our code under load, and see if our debugging operations break the driver.
- We should design our software for security - system users should not be able to break the driver with our debugging implementation.
- Think about field service engineers - debugging working hardware in the field.;
- Develop tools for service technicians to get debug information without powering down or dismantling the system.
- The Fedora distribution comes with debugfs already enabled.
- Android OS - the /dd directory gives debugfs information.
- Profiling the kernel - perf top, changes/effects in kernel
- The pokahole tool has nice way of looking at a static binary and listing how the data structures are laid out.
- dmidcode tool - looks at raw data, reports and displays things that it can recognize.
- For performance reasons, our debugging system should dump data from the driver in a binary format.
- Our implementation should work in two stages:
 - Collecting a snapshot of data in the driver
 - Displaying the data from the snapshot to the user
- Our tool should be able to work with whatever is dumped to debugfs (unrecognized snapshot, garbage, etc.)
- E1000 driver: supports ~60 devices.
- Ring structures in the E1000 driver: circular lists of descriptors.
- SPI information is available from a register.
- We must communicate with both sponsors, even if only one of them responds.
- Tuesday and Thursday at PSU around noon is the best time to meet on campus; come to their office at noon.
- Wednesday and Friday is best for afternoon meetings (Mondays not good for PJ)
- First meeting with sponsors at PSU will be Thursday, 5/3/12
- Shannon has done work on the concept that we will be implementing in our project, but would like to see someone else work on it.
- The sponsors would like to see a more generic version of their previous work.
- Tip: avoid printing specific info from the driver into a log.
- Tip: don't get scared about working inside of the driver or kernel.

- Tip: get a very simple first run of the project set up, with a custom compiled driver (with print statements), and a linux kernel with debugfs enabled.
 - Devin: more full team meetings would be a good idea throughout the project.
 - Summer term sponsor schedule: Monday and Tuesday afternoons on campus.
 - Sponsors can be on campus this term on Saturdays because they have a Saturday lab session for one of their courses.
 - PJ can bring his ThinkPad to help us with QEMU if necessary.
 - Devin: We should have a Saturday meeting in 2 weeks.
 - Final word from sponsors: don't hesitate to ask questions. There are no dumb questions.
 - Meeting with sponsors ended at 5:40 p.m.
-
- Brief team-only meeting begins.
 - Everyone should set up their own vm.
 - Use CapstoneCC box to set up our vm's.
 - Kyle will email script for vm setup.
 - David will follow script and compile his own copy of the custom kernel we will be using.
-
- Document deadlines:
 - Requirements document: first draft due 5/7, final draft due week 7
 - Work breakdown document: first draft due 5/14, final draft due week 7
 - Risk management document: final draft due week 7
 - V+V document:
 - Scheduling plan:
 - Team meeting next Wednesday, 5/9: we will break down project with the requirements known so far.
 - Bart has scheduled an all-hands meeting for Monday, 5/7.
 - Our team will hold a virtual meeting this Saturday.
 - Between now and the next team meeting: team members should be looking at the E1000 driver code, and looking for the adapter data structure in the driver code.
 - We need to make this project as simple as possible, with as few requirements as possible.
 - Devin wants to commit to very few requirements.
 - Team-only meeting ended at 6:05 p.m.