Monday, September 9

Review setup of QEMU and GeekOS

Run GeekOS again

Intro to/review of GDB

Tour of GeekOS source, with attention to Project 0

Discussion slides will be posted at http://www.cs.umd.edu/~jonfd/f13-412/
QEMU setup


Issues:

• sudo apt-get qemu... should have been sudo apt-get install qemu...

• You may also need to do sudo apt-get install nasm

• 64-bit OSs may cause problems
QEMU setup, cont.

Any other problems?

Mac users: VirtualBox seems to work fine for me; so does gdb 7.6.

If you are running in a VM, give it at least 1 GB of memory.

Linuxlab setup instructions on Wednesday.
QEMU/GeekOS demo

As described before, we run GeekOS with

    make run

in the build directory. All the programs in the user directory are available for you to run, though almost all of them will break for now.

However, we’d like a debugger. (Or: You will really want a debugger.)
Debugging GeekOS

This is an introduction; we will do a lot in the debugger as we go forward.

Use two terminal tabs. In one,

```
make dbgrun
```

In the other,

```
make dbg
```

Depending on your version of gdb, you may see a warning about auto-load safe-path. If you do, change the setting in ~/.gdbinit.
QEMU waits for gdb before doing anything. To start it,
  c (or continue)
Control-C in gdb to pause QEMU.
Breakpoints: restart, and before c, type
  break Sys_Spawn
which is the handler for the process creation syscall. Type c when done.
Using gdb

More instructions:
• step (s), next (n), finish
• print (p), examine (x)

Other breakpoint specs:
• break file.c:line

Help:
• help [topic or keyword]
• apropos topic
• http://users.ece.utexas.edu/~adnan/gdb-refcard.pdf
GeekOS Source

syscall.c: entry point for system call handlers
pipe.c: where you will do most of Project 0
vfs.c: indirection layer for Read, Write, Close
More to come...