Project 2b: Scheduling and Virtual Memory in xv6

Important Dates

Questions about the project? Use piazza.

Due: Monday 10/24. However, 2 late days available for free (grace period).

Updates

Overview

IMPORTANT: Read this ENTIRE PAGE as there are differences from the github project!

The scheduling part of the project is described here. Please read the description carefully and watch this video.

The virtual memory part of the project is described here. Please read the description carefully and watch this video.

Differences

PLEASE READ: There are some differences in this project versus the descriptions above. Notably:

- Your scheduler will not be a lottery scheduler; it will instead be a simple priority-based scheduler. Specifically, if a process sets its ticket value to 1 it has **high** priority; otherwise, a process should have a ticket value of 0 and thus have **low** priority. All other ticket values are not valid; 1 (high priority) should be the default. High-priority processes should always have priority over low-priority processes, e.g., if there is a single high-priority process and a single low-priority process, the high-priority one will run to completion (thus starving the low-priority process). If there two or more high-priority processes, they should alternate round-robin style; similarly, if there are no high-priority processes and more than one low-priority processes, they should alternate round-robin.
- The graph should just show that your priority scheduler works; for example, imagine two processes running with high priority, then one lowers its priority, then the other runs to completion, then finally the low priority process runs to completion, or something like that.

Notes

This project is the last one to be **done by yourself** this semester. Congratulations on making it this far! Copying code is considered cheating. Read this for more info on what is OK and what is not.

This project is to be done on the lab machines (listed here), so you can learn more about programming in C on a typical UNIX-based platform (Linux).

The Code

The source code for xv6 (and associated README) can be found in **~cs537-1/public/xv6.tgz** . Everything you need to build and run and even debug the kernel is in there.

Handing It In

The handin directory is `~cs537-1/handin/login/p2b` where `login` is your login.

Copy all of your source files (but not .o files, please, or binaries!) into the `p2b/` subdirectory. A simple way to do this is to copy everything into the destination directory directory, then type `make` to make sure it builds, and then type `make clean` to remove unneeded files.

Finally, in your p2b directory, please make a README file to describe what you did, and include your graph in there too, called **graph.pdf** .