Project 2: Processes and Scheduling

Due: ~~Sunday, July 9th~~ Tuesday, July 11th before 11:59 pm

You may optionally work with a project partner for this project.

Questions?

Please post your questions on Piazza so that your classmates with similar problems can benefit from it. However, if the question is about your code, visiting during **office hours** is the most efficient way of getting help. You should **not** post your code on Piazza.

Overview

There are two parts to this project:

- Shell: to be done on a linux machine, so you can learn more about programming in C on a typical UNIX-based platform (Linux). Before you begin, we highly recommend that you watch this video of a previous discussion section; note that the project specification has changed in some ways from this previous discussion section, but all of the discussion in the video remains relevant.
- Scheduling: to be done in our xv6 OS hacking environment. Again, before you begin this part, carefully watch this video. You may want to watch this one several times! Again, you aren't implementing the exact same scheduler, but you still need to understand how the context switching code works.

Click on the above links to learn more about what you should do. READ EACH CAREFULLY!

Handing It In

If you are working in pairs, only one of you need to submit the source code.

Your handin directory is `~cs537-1/handin/`**`$USER`**`/p2` where $USER is your CS login. Copying of these files is accomplished with the `cp` program, as follows:

```
shell% cp xsh.c ~cs537-1/handin/$USER/p2/linux/
```

For the C/Linux part of this project (the shell), you should turn the .c files you created. You should copy these .c files into the subdirectory called `linux` within your handin directory for p2. You should also include a simple **Makefile**, so we can simply type `make` and build your testable binary. The binary should be named `xsh` .

For the xv6 part of the project, copy all of your source files (but not .o files, please, or binaries!) into the `xv6/` subdirectory of your p2 directory. A simple way to do this is to copy everything into the destination directory, then type `make` to make sure it builds, and then type `make clean` to remove unneeded binary files.

Into your p2 directory, put a `README` file. Describe a little bit about what you did for this project in

this file.

Finally, also most importantly, if you are working with a partner, create another file called `partner.txt` in your p2 directory, where you should **only write your partner's CS login**. Both of you should create this file to be graded, and again, only one of you need to submit the source files. If there is no such file, we are assuming you are working alone.

For example, if Xiangjin works with Zhewen and let's assume Xiangjin is submitting the source files, then Xiangin will be turning in the source + the `partner.txt` file and Zhewen will be turning in only the `partner.txt` file. Xiangjin's `partner.txt` will contain a single line of Zhewen's CS login and Zhewen's `partner.txt` will contain only a single line of Xiangjin's CS login.