

Lab 1: Introduction to xv6

Please replace red text with your report text and any tables or figures, names of any accompanying files, etc. Remember to commit all the files for your lab submission, to put the URL for your private xv6 repo in the Teams assignment, to submit the Teams assignment, and to give the instructor, TA, and IA access to your repo.

Task 1. Boot xv6 and explore utilities

Describe your Linux VM setup (e.g., what hardware, what VMM did you use, which Linux flavor and version?)

Show the result of building and booting xv6 on (emulated) RISC-V. Show the results of ls. Give the results from the three other commands you explored and explain how they work.

Describe any difficulties you ran into and if/how you overcame them.

I was having issues getting a vm setup, I think my hardware either wasn't compatible or didn't support the vm. I which is weird since I was trying to use the same one that I used for computer arch. Eventually, I decided to go the Linode route and I was able to get a vm setup and ssh. Currently I'm using vim to change my files, I'm still trying to get used to it. I'm not sure if I like it yet.

```

[user@Hernans-Mac ~ % ssh hernan@45.33.101.85
[hernan@45.33.101.85's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Sep 11 12:25:48 AM UTC 2022

System load:          0.0
Usage of /:           10.4% of 78.17GB
Memory usage:         6%
Swap usage:           0%
Processes:            120
Users logged in:      0
IPv4 address for eth0: 45.33.101.85
IPv6 address for eth0: 2600:3c02::f03c:93ff:fe5e:c969

14 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Sat Sep 10 18:15:11 2022 from 99.118.68.186
[hernan@localhost:~$ cd Documents/OS/xv6-riscv-labs/
hernan@localhost:~/Documents/OS/xv6-riscv-labs$ >

[hernan@localhost:~/Documents/OS/xv6-riscv-labs$ make qemu
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -
o-mmio-bus.0

xv6 kernel is booting

hart 1 starting
hart 2 starting
init: starting sh
$ █

```

Task 2.

Your code `sleep.c` should be in the user directory in the lab1 branch of your xv6 repo.

Summarize what you learned by carrying out this task.

Describe any difficulties you ran into with this task and if/how you overcame them.

I did pretty much what it said in one of the slides. I copied the `kill.c` file and changed `kill()` to `sleep()`

Task 3.

List all the files you added or modified to do this task, along with the purpose of each addition or modification. For example

user/user.h – added getprocs() prototype: int getprocs(struct uproc*)

user/ps.c – implementation of the ps user command

.

.

kernel/sysproc.c – added sys_getprocs() function that xxxxxx (describe what it does for xxxxxx)

Show the output of your ps command, including while running your ptest code in the background.

Summarize what you learned by carrying out this task (should be several things).

Describe any difficulties you ran into with this task and if/how you overcame them.

This task was especially difficult. It took me a full day to get it work. The homework helped with getting everything setup, however making sure that my logic was working was a challenge, especially in the procinfo function in the proc.c file. There was a time where I wasn't getting the desired output and I thought it was my copyout, however that wasn't the case. The issue was with my getproc function under sysproc.c. I was able to solve it by reading over the documents provided by Dr. Moore and seeing what I had to model my files after. I think that was the most challenging thing, finding out why my functions were working, yet not passing the right info to eachother.

Task 4.

Your code ptree.c and your test case code ptest.c should be in the user directory.

Show the results of running and testing your ptree command.

Describe any difficulties you ran into with this task and if/how you overcame them.

I'm not sure if this was the right approach by I did change the mktree function parameters a bit in order to make it easier to me. I passed the parent id in order to keep track of the indentation of each process. I think it might not have been necessary but this was the easiest way for me to visualize it.

Discussion Question: Answer the discussion question as well as you can, given what you have learned at this point in the class.

We have not discussed concurrency control and locking yet. Given what you know so far about xv6, what could possibly go wrong with ps or ptree with no concurrency control being used?

A: I think without concurrency the process will change if they are called from different terminals or if they are called after system call sleep.