CS4375-13948 Fall 2023 Homework Report

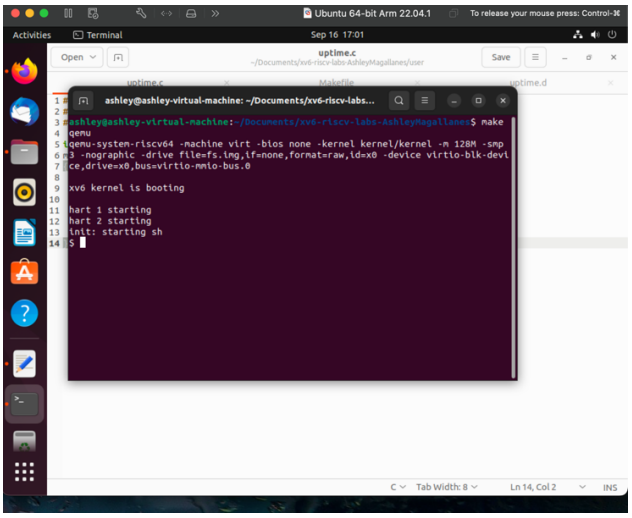
Ashley Magallanes /Submitted on Sep 16, 2023

avmagallanes@utep.miners.edu

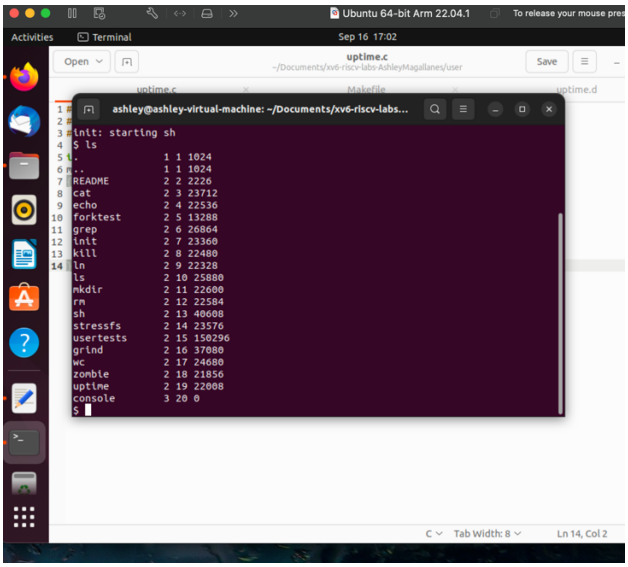
HW 1: Introduction to xv6

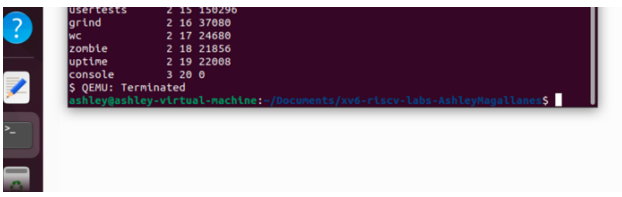
Task 1. Boot xv6 and explore utilities

To boot xv6 I obtained Ubuntu server operating system I had to download a virtual machine along with ubuntu itself. I did this through VMware Fusion virtual machine. This allowed me to obtain Ubuntu 64-bit 22.04.1 on my Apple M2 MacBook. After duplicating Dr.Moore's repository and setting up my git terminal in Ubuntu, I used the command "make qemu" in order to boot the system. The following image showcases the xv6 kernel booting system.

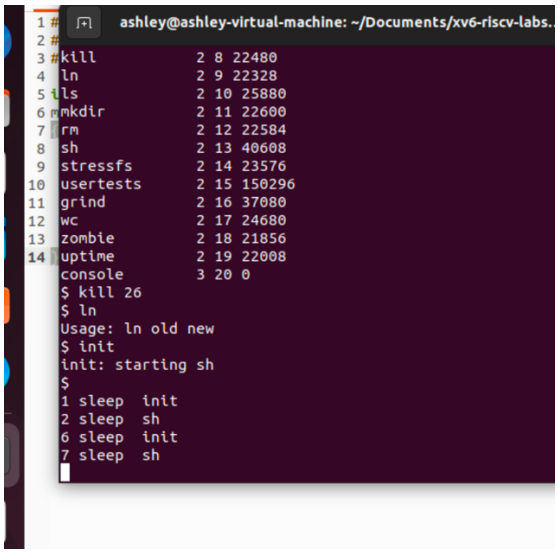


When running the 'ls' process, all the initial files in the system are showcased. This is demonstrated in the following screenshot.



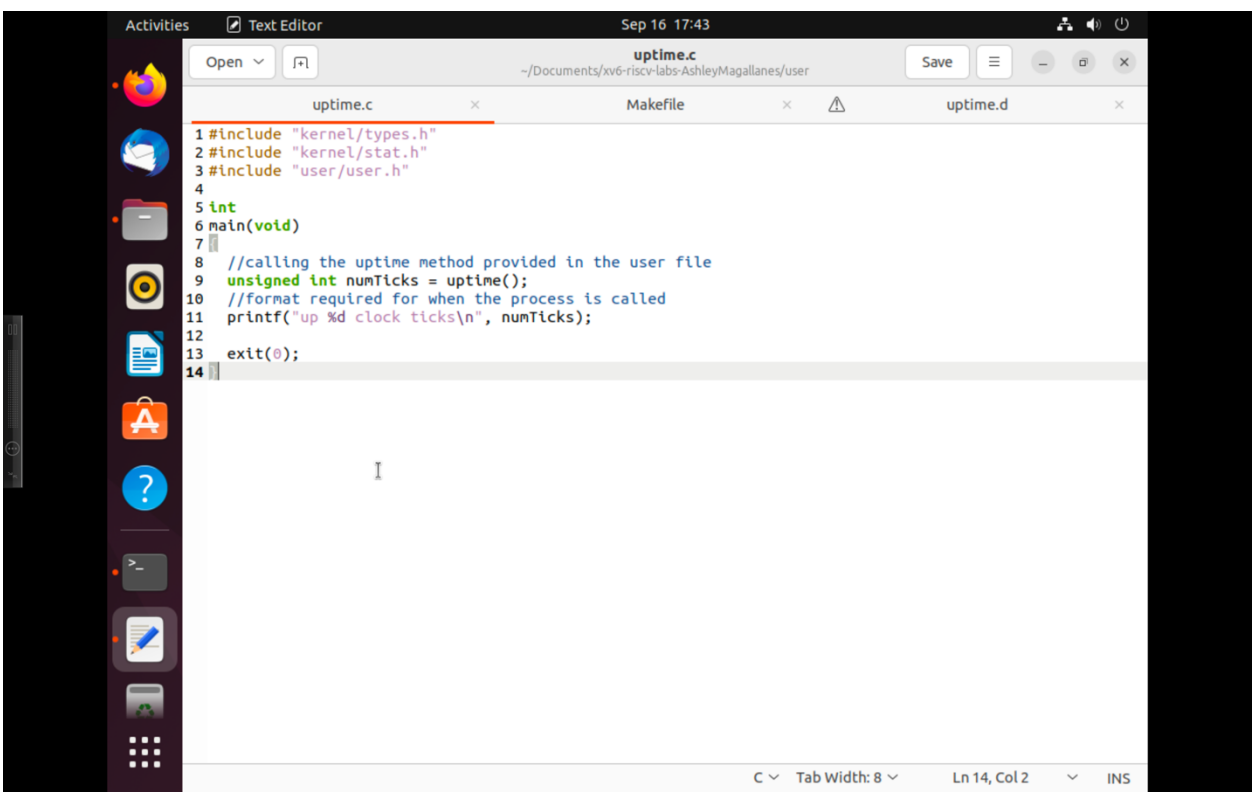


I tried the commands specified on the assignment document, including the command to terminate qemu. In addition to the 'ls' command I tried the commands: kill, ln, and init. The 'kill' command just manages to terminate the program which the user wants to end. This process takes an argument which is the process identification number of the program that will be terminated. Then it checks if the process id used as the argument is less than two, and if it is the process prints "usage: kill pid…." Then if the process id is not less than two the kill method is called and the process is terminated. The 'ln' command showcases "Usage: In old new", this command is used to create files. Lastly the "init" command starts the a process, this is demonstrated by the "sh" symbol. Then I also tried the ctlr-p command which produced the sleep schedule for all the processes I did before.

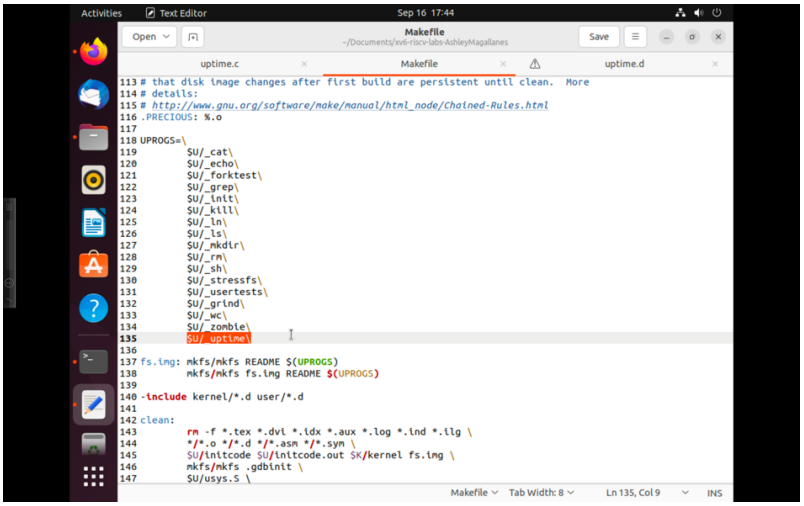


Other than downloading the virtual machine in my M2 MacBook and then setting up the repository, I had a hard time understanding how qemu worked. Since I had never seen it before I was confused. However, after trial-and-error experience I managed to understand it better. Another thing that I had a hard time doing was killing some processes that were not letting my git be installed on the Ubuntu terminal.

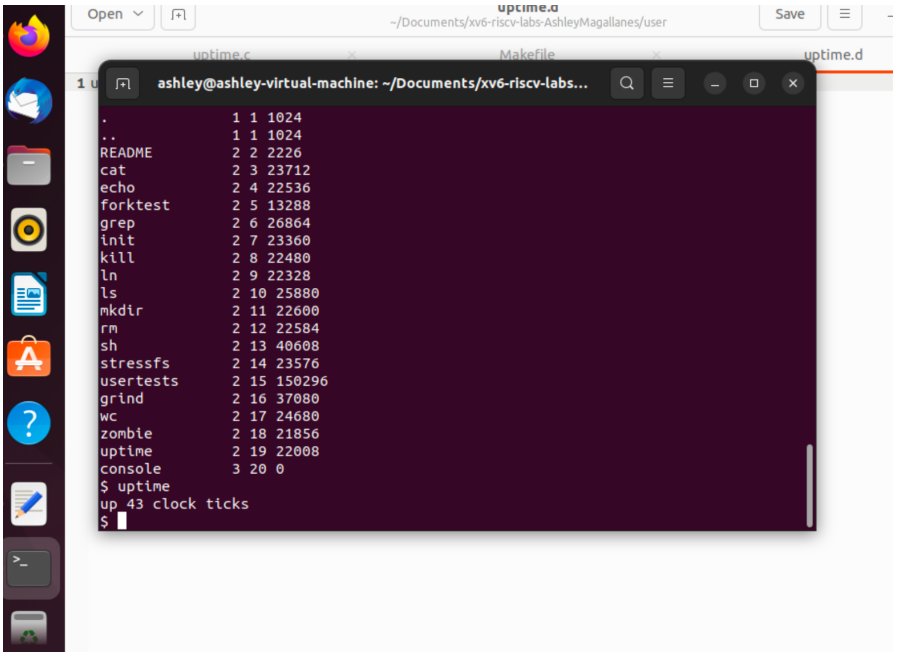
Task 2. Implement the uptime utility



I Implemented the UNIX utility uptime for xv6 by adding a C file to the user folder on the repository I duplicated. This file calls the method uptime() which was already provided by Dr.Moore, then it prints the outcome in the correct format. This format was not only the one wanted by the assignment instructions, but I tried the “uptime” command on my own MacBook terminal and the format is generally universal. That is something nice to learn.



In order for my new uptime.c file to be included in the booting system I had to add it to the MakeFile already set up. I just added the ‘$U/_uptime\’ line which allows the program to also consider uptime as a UPROG. Lastly, I noticed that all of the other command files like kill and init had a ‘.d’ file corresponding to it. Therefore, I also created one for uptime. Saved everything and run ‘qemu’ again. Then I tested it worked and the following image illustrates the successful outcome. Resulting in the uptime command showcasing the ‘up 43 clock ticks.’ Overall, this task was not too confusing and it was easy to follow.


Open
uptime.d
~/Documents/xv6-riscv-labs-AshleyMagallanes/user
Save
uptime.c
Makefile
uptime.d
ashley@ashley-virtual-machine: ~/Documents/xv6-riscv-labs...
. 1 1 1024
.. 1 1 1024
README 2 2 2226
cat 2 3 23712
echo 2 4 22536
forktest 2 5 13288
grep 2 6 26864
init 2 7 23360
kill 2 8 22480
ln 2 9 22328
ls 2 10 25880
mkdir 2 11 22600
rm 2 12 22584
sh 2 13 40608
stressfs 2 14 23576
usertests 2 15 150296
grind 2 16 37080
wc 2 17 24680
zombie 2 18 21856
uptime 2 19 22008
console 3 20 0
$ uptime
up 43 clock ticks
$