

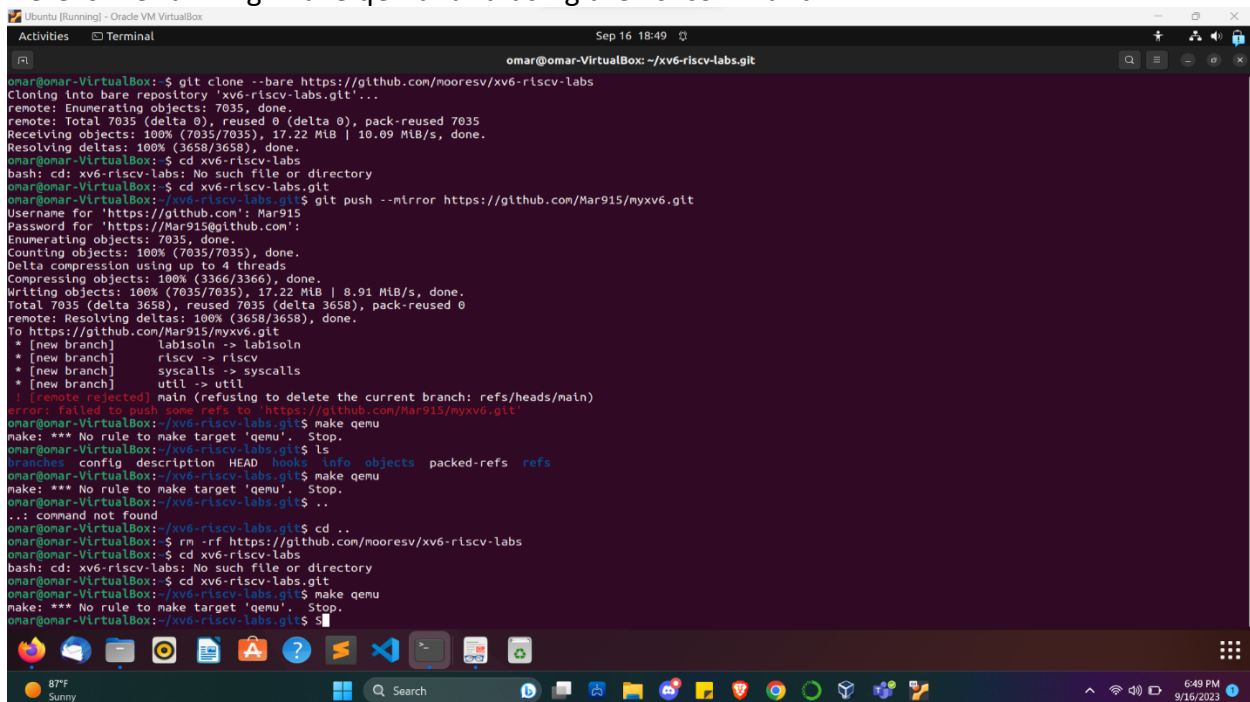
HW 1: Introduction to xv6

<https://github.com/Mar915/myxv6.git>

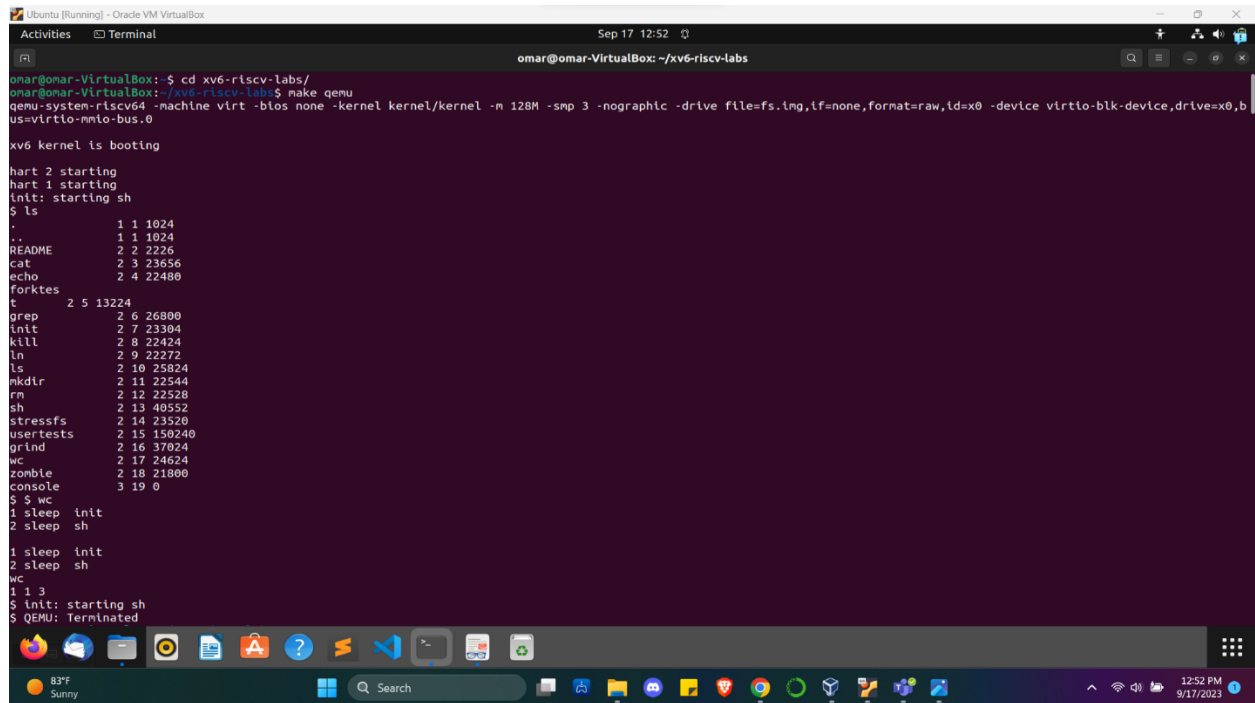
Task 1. Boot xv6 and explore utilities

I used the virtual machine “Oracle VM Virtualbox” and used Ubuntu 22.04 LTS (Jammy Jellyfish).

Here is me running ‘make qemu’ and using the ‘ls’ command:



```
ommar@ommar-VirtualBox: ~/xv6-riscv-labs.git
ommar@ommar-VirtualBox:~$ git clone --bare https://github.com/mooresv/xv6-riscv-labs
Cloning into bare repository 'xv6-riscv-labs.git'...
remote: Enumerating objects: 7035, done.
remote: Total 7035 (delta 0), reused 0 (delta 0), pack-reused 7035
Receiving objects: 100% (7035/7035), 17.22 MiB | 10.09 MiB/s, done.
Resolving deltas: 100% (3658/3658), done.
ommar@ommar-VirtualBox:~$ cd xv6-riscv-labs
bash: cd: xv6-riscv-labs: No such file or directory
ommar@ommar-VirtualBox:~$ cd xv6-riscv-labs.git
ommar@ommar-VirtualBox:~/xv6-riscv-labs.git$ git push --mirror https://github.com/Mar915/myxv6.git
Username for 'https://github.com': Mar915
Password for 'https://github.com': 
Enumerating objects: 7035, done.
Counting objects: 100% (7035/7035), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3366/3366), done.
Writing objects: 100% (7035/7035), 17.22 MiB | 8.91 MiB/s, done.
Total 7035 (delta 3658), reused 7035 (delta 3658), pack-reused 0
remote: Resolving deltas: 100% (3658/3658), done.
To https://github.com/Mar915/myxv6.git
 * [new branch]      labisoln -> labisoln
 * [new branch]      riscv -> riscv
 * [new branch]      syscalls -> syscalls
 * [new branch]      util -> util
! [remote rejected] main (refusing to delete the current branch: refs/heads/main)
error: failed to push some refs to 'https://github.com/Mar915/myxv6.git'
ommar@ommar-VirtualBox:~/xv6-riscv-labs.git$ make qemu
make: *** No rule to make target 'qemu'.  Stop.
ommar@ommar-VirtualBox:~/xv6-riscv-labs.git$ ls
branches  config  description  HEAD  hooks  info  objects  packed-refs  refs
ommar@ommar-VirtualBox:~/xv6-riscv-labs.git$ make qemu
make: *** No rule to make target 'qemu'.  Stop.
ommar@ommar-VirtualBox:~/xv6-riscv-labs.git$ ..
.: command not found
ommar@ommar-VirtualBox:~/xv6-riscv-labs.git$ cd ..
ommar@ommar-VirtualBox:~$ rm -rf https://github.com/mooresv/xv6-riscv-labs
ommar@ommar-VirtualBox:~$ cd xv6-riscv-labs
bash: cd: xv6-riscv-labs: No such file or directory
ommar@ommar-VirtualBox:~$ cd xv6-riscv-labs.git
ommar@ommar-VirtualBox:~/xv6-riscv-labs.git$ make qemu
make: *** No rule to make target 'qemu'.  Stop.
ommar@ommar-VirtualBox:~/xv6-riscv-labs.git$
```



```
ommar@ommar-VirtualBox:~$ cd xv6-riscv-labs/
ommar@ommar-VirtualBox:~/xv6-riscv-labs$ make qemu
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -drive file=fs.img,if=none,format=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0

xv6 kernel is booting

hart 2 starting
hart 1 starting
init: starting sh
$ ls
.          1 1 1024
..         1 1 1024
README    2 2 2226
cat       2 3 23656
echo      2 4 22480
forktest  2 5 13224
t         2 6 26800
grep      2 7 23394
init      2 8 22424
kill      2 9 22272
ln        2 10 25824
mkdir     2 11 22544
rm        2 12 22528
sh        2 13 40552
stressfs  2 14 23520
usertests 2 15 150240
grind     2 16 37024
wc        2 17 24624
zombie    2 18 21800
console   3 19 0
$ wc
1 sleep  init
2 sleep  sh

1 sleep  init
2 sleep  sh
wc
1 1 3
$ init: starting sh
$ QEMU: Terminated
```

- **Echo**

Prints messages or text on the terminal console.



```
$ echo Hello CS4375
Hello CS4375
$
```

- **Wc**

Stands for “word count” and it counts the number of words, lines, and characters in a text file.



```
$ wc README
47 317 2226 README
$
```

- **Cat**

Concatenates and displays the contents of one or more text files.

```
$ cat README
xv6 is a re-implementation of Dennis Ritchie's and Ken Thompson's Unix
Version 6 (v6). xv6 loosely follows the structure and style of v6,
but is implemented for a modern RISC-V multiprocessor using ANSI C.

ACKNOWLEDGMENTS

xv6 is inspired by John Lions's Commentary on UNIX 6th Edition (Peer
to Peer Communications; ISBN: 1-57398-013-7; 1st edition (June 14,
2000)). See also https://pdos.csail.mit.edu/6.828/, which
provides pointers to on-line resources for v6.

The following people have made contributions: Russ Cox (context switching,
locking), Cliff Frey (MP), Xiao Yu (MP), Nickolai Zeldovich, and Austin
Clements.

We are also grateful for the bug reports and patches contributed by
Takahiro Aoyagi, Silas Boyd-Wickizer, Anton Burtsev, Ian Chen, Dan
Cross, Cody Cutler, Mike Cai, Tej Chajed, Asant Dol, eval2800, Nelson
Elhage, Saar Ettlinger, Alice Ferrazzi, Nathaniel Filardo, flespark,
Peter Froehlich, Yakir Gao, Shivan Handa, Matt Harvey, Bryan Henry,
jaichenhengjie, Jim Huang, Matúš Jókay, Alexander Kapshuk, Anders
Kaseorg, kehao95, Wolfgang Keller, Jungwoo Kim, Jonathan Kimmitt,
Eddie Kohler, Vadim Kolontsov, Austin Liew, löstman, Pavan
Maddamssetti, Inbar Marinescu, Yandong Mao, , Matan Shabtay, Hitoshi
Mitake, Carmi Morimovich, Mark Morrissey, ntasn, Joel Nider,
OptimisticSide, Greg Price, Jude Rich, Ayan Shafqat, Eldar Sehayek,
Yongming Shen, Fumiya Shigemitsu, Cam Tenny, tyfkda, Warren Toomey,
Stephen Tu, Rafael Ubal, Amane Uehara, Pablo Ventura, Xi Wang, Keiichi
Matanabe, Nicolas Molovick, wxdao, Grant Wu, Jindong Zhang, Icenowy
Zheng, ZhuYu1997, and Zou Chang Mei.

The code in the files that constitute xv6 is
Copyright 2006-2020 Frans Kaashoek, Robert Morris, and Russ Cox.

ERROR REPORTS

Please send errors and suggestions to Frans Kaashoek and Robert Morris
(kaashoek,rtm@mit.edu). The main purpose of xv6 is as a teaching
operation system for MIT's 6.S081, so we are more interested in
```

The difficulties I had was cloning and mirroring the repositories. How I overcame it was asking the TA for help and both of us troubleshooting the problem.

Task 2. Implement the uptime utility

My main issue was getting the C compiler to run my uptime.c through the Makefile. It would not create an executable for some reason which in turn would error out the whole xv6 system. My solution to that was running uptime.c in my desktop with a simple "Hello World" printing statement in order to get the executable to be produced. I then added the C file and executable and was able to run the Makefile with no problem.