

## HW 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 and TA access to your repo.

### Task 1. Boot xv6 and explore utilities

I used the Windows Subsystem for Linux- WSL 2

```
ke: *** No rule to make target 'qemu'. Stop.
tshering@DESKTOP-208BNCM1:~$ sudo apt-get install git build-essential gdb-multiarch qemu-system-misc gcc-riscv64-linux-gnu binutils-riscv64-linux-gnu
[sudo] password for sstshering:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9ubuntu3).
gcc-riscv64-linux-gnu is already the newest version (4:11.2.0-1ubuntu1).
binutils-riscv64-linux-gnu is already the newest version (2.38-4ubuntu2.3).
gdb-multiarch is already the newest version (1:12.34.1-1ubuntu1.10).
qemu-system-misc is already the newest version (1:6.2+dfsg-2ubuntu6.13).
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 91 not upgraded.
tshering@DESKTOP-208BNCM1:~$ git clone --bare https://github.com/mooresv/xv6-riscv-labs.git
Cloning into 'xv6-riscv-labs.git'...
remote: Enumerating objects: 7035, done.
remote: Counting objects: 100% (7035/7035), done.
remote: Compressing objects: 100% (3364/3364), done.
remote: Total 7035 (delta 3660), reused 7035 (delta 3660), pack-reused 0
Receiving objects: 100% (7035/7035), 17.22 MiB | 1.50 MiB/s, done.
Resolving deltas: 100% (3660/3660), done.
tshering@DESKTOP-208BNCM1:~$ cd OS-riscv-xv6.git
bash: cd: OS-riscv-xv6.git: No such file or directory
tshering@DESKTOP-208BNCM1:~$ cd xv6-riscv-labs.git
bash: cd: xv6-riscv-labs.git: No such file or directory
Select sstshering@DESKTOP-208BNCM1: ~ / OS-riscv-xv6
sstshering@DESKTOP-208BNCM1:/home$ ls
sstshering
sstshering@DESKTOP-208BNCM1:/home$ cd sstshering/
sstshering@DESKTOP-208BNCM1:~/sstshering$ git clone https://github.com/sstshering/OS-riscv-xv6.git
Cloning into 'OS-riscv-xv6'...
remote: Enumerating objects: 7035, done.
remote: Counting objects: 100% (7035/7035), done.
remote: Compressing objects: 100% (3364/3364), done.
remote: Total 7035 (delta 3660), reused 7035 (delta 3660), pack-reused 0
Receiving objects: 100% (7035/7035), 17.22 MiB | 1.50 MiB/s, done.
Resolving deltas: 100% (3660/3660), done.
sstshering@DESKTOP-208BNCM1:~/OS-riscv-xv6$ cd OS-riscv-xv6/
sstshering@DESKTOP-208BNCM1:~/OS-riscv-xv6$ ls
LICENSE Makefile README kernel mkfs user
sstshering@DESKTOP-208BNCM1:~/OS-riscv-xv6$ make qemu
riscv64-linux-gnu-gcc -c -o kernel/entry.o kernel/entry.S
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/start.o kernel/start.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/console.o kernel/console.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/printk.o kernel/printk.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/uart.o kernel/uart.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/kalloc.o kernel/kalloc.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/spinlock.o kernel/spinlock.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/string.o kernel/string.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/main.o kernel/main.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/vm.o kernel/vm.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/proc.o kernel/proc.c
riscv64-linux-gnu-gcc -c -o kernel/switch.o kernel/switch.S
riscv64-linux-gnu-gcc -c -o kernel/trampoline.o kernel/trampoline.S
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/trap.o kernel/trap.c
```

```
Select sstshering@DESKTOP-20BNCM1: ~/OS-riscv-xv6
mkfs/mkfs fs.img README user_cat user_echo user_forktest user_grep user_init user_kill user_ln user_ls user_mkdir user_rm user_sh user_stressfs user_usertests
user_grind user_wc user_zombie user_sleep user_ps user_pstree user_pstest
mmeta 46 (boot, super, log blocks 30 inode blocks 13, bitmap blocks 1) blocks 954 total 1000
ballocc: first 683 blocks have been allocated
ballocc: write bitmap block at sector 45
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 1 starting
hart 2 starting
init: starting sh
$ ls
.          1 1 1024
..         1 1 1024
README    2 2 2226
cat        2 3 23720
echo       2 4 22560
forktest  2 5 13280
grep       2 6 26864
init       2 7 23360
kill       2 8 22480
ln         2 9 22352
ls         2 10 25904
mkdir      2 11 22616
rm         2 12 22600
sh         2 13 40624
stressfs   2 14 23576
usertests  2 15 150312
grind      2 16 37096
wc         2 17 24696
zombie     2 18 21856
sleep      2 19 22772
ps         2 20 23400
pstree     2 21 24360
pstest     2 22 23456
console    3 23 0
$
$ ls
```

I explored echo, cat and mkdir commands.

**echo** is used to print message to the terminal. It takes the text use types in and prints it to the terminal.

**cat** is used to concatenate and display the contents of the file(s). It reads the contents of the mentioned file by user and displays it to the terminal. For example, I did cat README file.

**mkdir** is used to create a new directory/folder. It follows the directory's name and creates it if there's no pre-existing directory.

```
Select sstshering@DESKTOP-20BNCM1: ~/OS-riscv-xv6
console    3 23 0
$ echo Hello
Hello
$ 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 CAT, Tej Chajed, Asami Doi, eyalz800, Nelson
Elhage, Sear Ettinger, Alice Ferrazzi, Nathaniel Filardo, flespark,
Peter Froehlich, Yakir Goaron, 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, l0stman, Pavan
Maddamsetti, Imbar Marinescu, Yandong Mao, , Matan Shabtay, Hitoshi
Mitake, Carmi Merimovich, Mark Morrissey, mtasm, 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
Watanabe, Nicolas Wolovick, wxdao, Grant Wu, Jindong Zhang, Icenowy
Zheng, ZhuYU1997, and Zou Chang Wei.

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

ERROR REPORTS
```

```
Select sstshering@DESKTOP-20BNM1: ~/OS-riscv-xv6
BUILDING AND RUNNING XV6

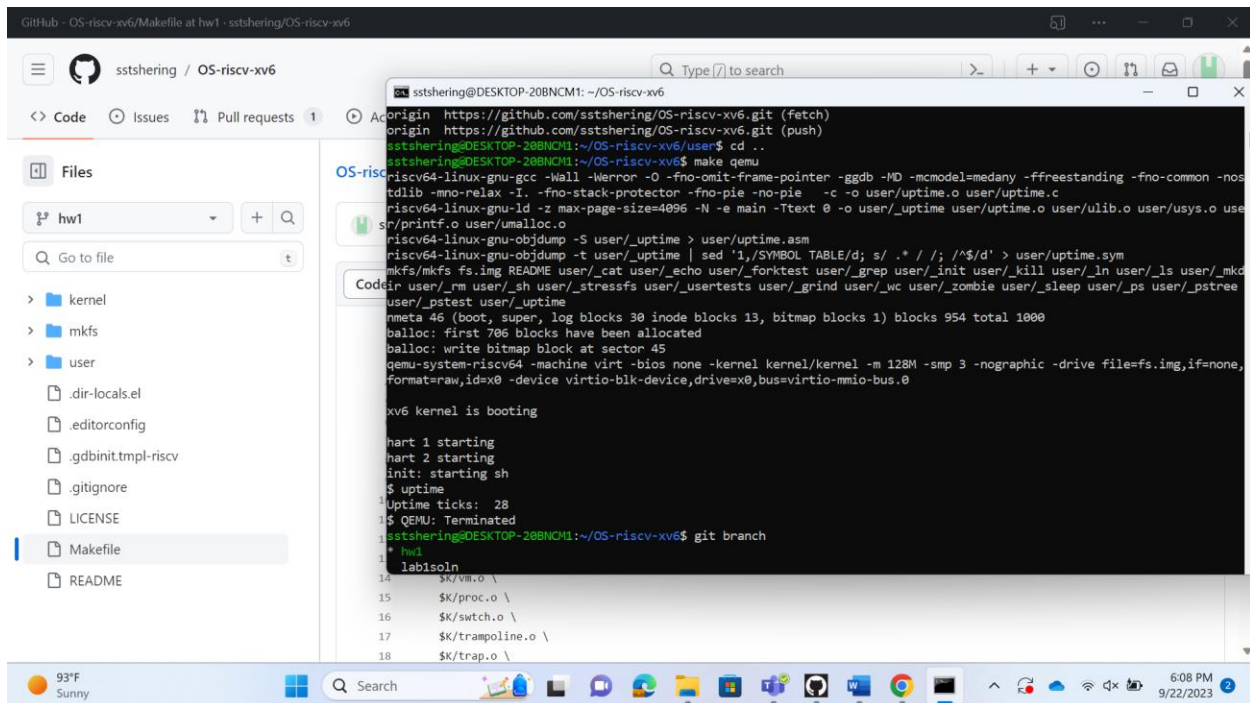
You will need a RISC-V "newlib" tool chain from
https://github.com/riscv/riscv-gnu-toolchain, and qemu compiled for
riscv64-softmmu. Once they are installed, and in your shell
search path, you can run "make qemu".
$ mkdir hw1
$ ls
.          1 1 1024
..         1 1 1024
README    2 2 2226
cat        2 3 23720
echo       2 4 22560
forktest  2 5 13280
grep       2 6 26864
init       2 7 23360
kill       2 8 22480
ln         2 9 22352
ls         2 10 25904
mkdir      2 11 22616
rm         2 12 22600
sh         2 13 40624
stressfs   2 14 23576
usertests  2 15 150312
grind      2 16 37096
wc         2 17 24696
zombie     2 18 21856
sleep      2 19 22272
ps         2 20 23400
pstree     2 21 24360
pstest     2 22 23456
console    3 23 0
hw1        1 24 32
$ cd hw1
$ ls
exec ls failed
$
```

The most difficulties I faced was cloning the repo and getting it onto my ubuntu.

## Task 2. Implement the uptime utility

```
AutoSave OFF CS4375 Fall2023 HW1 Report template 2 No Label Search Tshering, Sonam S 15
File Home Insert Draw Design Layout References Mailings Review View Help Comments Editing Share
Clipboard Font
Task 2
sstshering@DESKTOP-20BNM1: ~/OS-riscv-xv6/user
forktest _wc grep.c initcode.d ls.sym pstest.o sleep.asm uptime.c wc.o
_grep _zombie grep.d initcode.o mkdir.asm pstest.sym sleep.c uptime.d wc.sym
_grind cat.asm grep.o initcode.out mkdir.c pstree.asm sleep.d uptime.o zombie.asm
_init cat.c grep.sym kill.asm mkdir.d pstree.c sleep.o uptime.sym zombie.c
_kill cat.d grind.asm kill.c mkdir.o pstree.d sleep.sym user.h zombie.d
_ln cat.sym grind.c kill.o mkdir.sym pstree.o stressfs.asm usertests.asm zombie.o
_ls cat.o grind.d kill.o printf.c pstree.sym stressfs.c usertests.c zombie.o
mkdir echo.asm grind.o kill.o printf.d rm.asm stressfs.d usertests.o
ps echo.c grind.sym ln.asm printf.o rm.c stressfs.o usertests.o
pstest echo.d init.asm ln.c ps.asm rm.o ulib.c usys.S
_rm echo.o init.c ln.d ps.c rm.o ulib.d usys.d
sh forktest.asm init.o ln.sym ps.o sh.asm ulib.o usys.o
sleep forktest.c init.sym ls.asm ps.sym sh.c umalloc.c usys.pl
stressfs forktest.d initcode ls.c pstest.asm sh.d umalloc.d wc.asm
sstshering@DESKTOP-20BNM1:~/OS-riscv-xv6/user$ git branch
* hw1
lab1soln
sstshering@DESKTOP-20BNM1:~/OS-riscv-xv6/user$ cat uptime.c
#include "kernel/types.h"
#include "kernel/stat.h"
#include "user/user.h"

int main(void){
    int uptimeTicks = uptime();
    printf("Uptime ticks: %d\n", uptimeTicks);
    exit(0);
}
```



I learned mainly how to git pull, push and commit between local and remote repo. Besides that, I learned about how to implement uptime on xv6.

I was able to push my hw1 branch to the origin but the new updates I did on here weren't showing up on the local repository. But everything ran, I was able to run it on xv6 shell.

