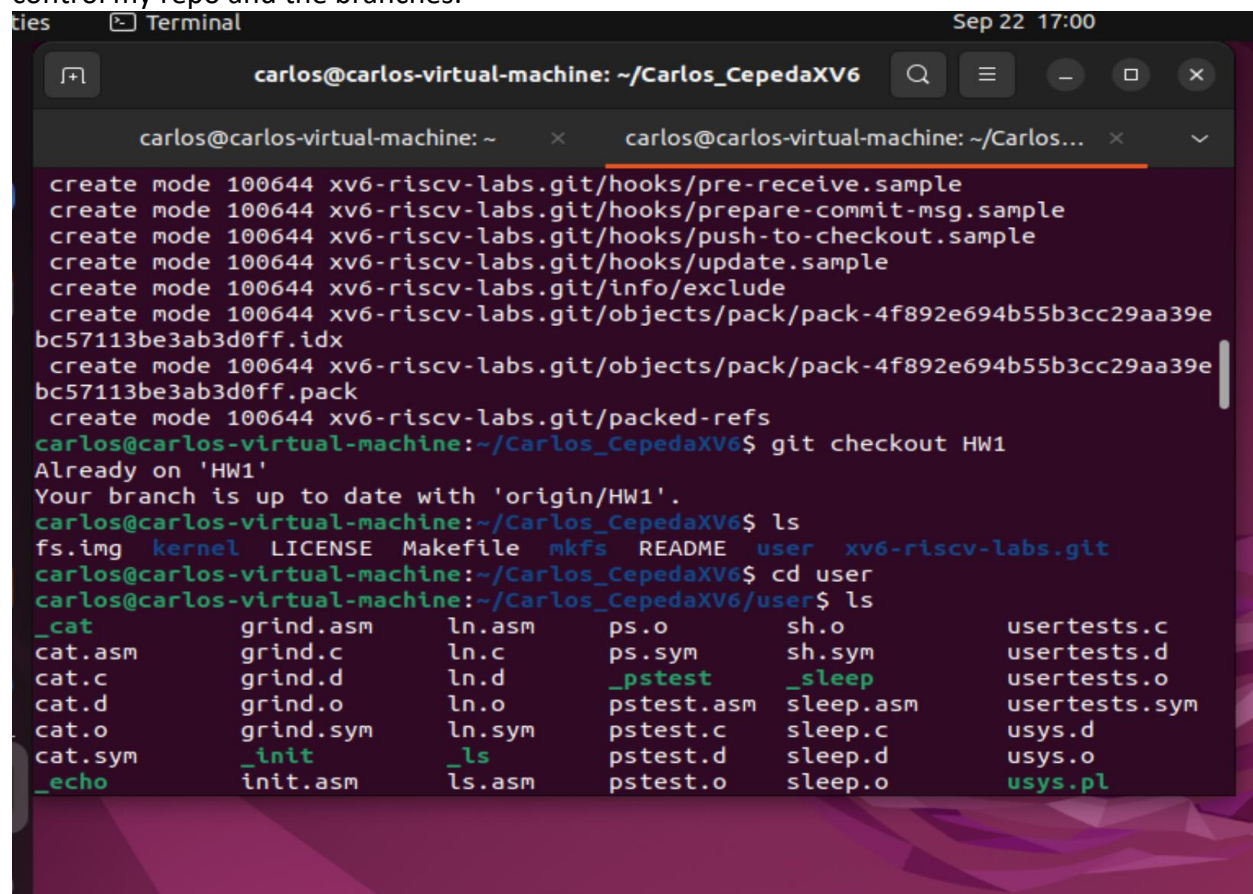


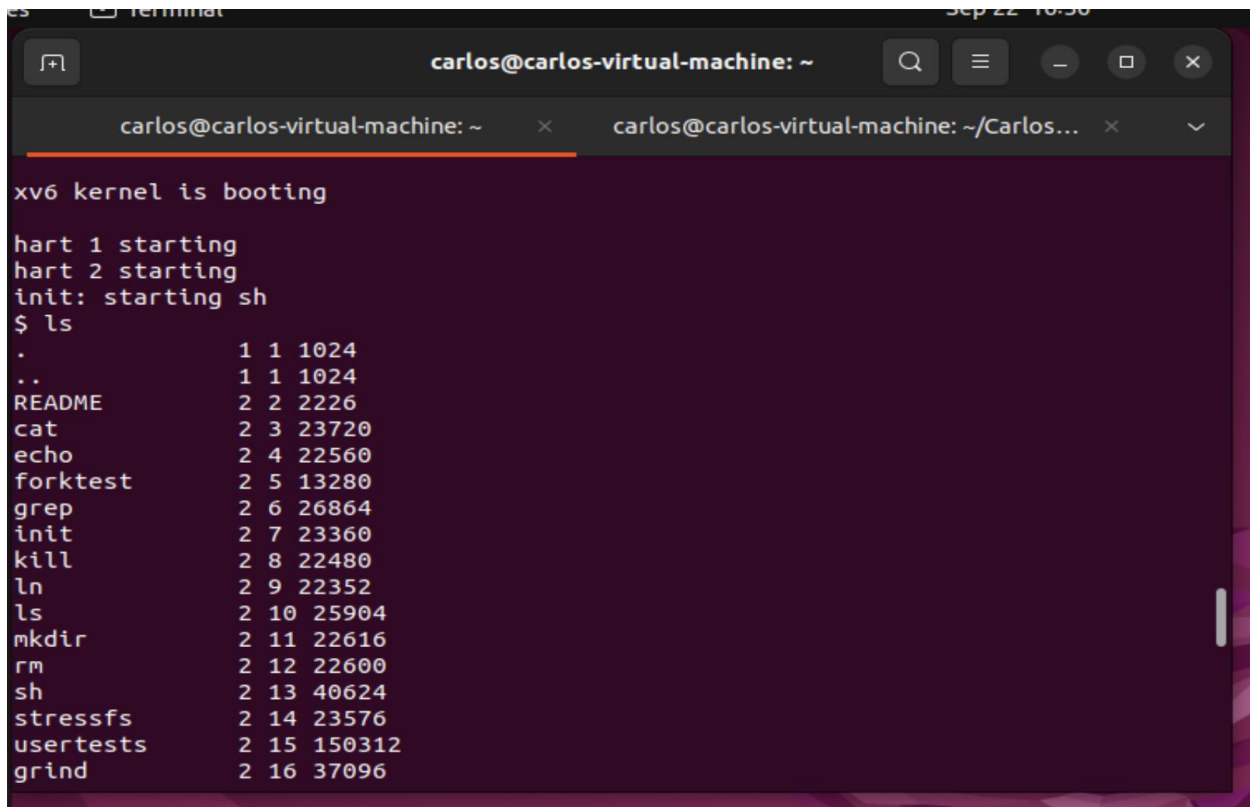
HW 1: Introduction to xv6

Task 1. Boot xv6 and explore utilities

I'm using VMware workstation pro, and installed ubuntu into it. It took a long time to set up since I don't have a very powerful laptop. I started playing to remember how to move around bash. Using ls to get see all the files, cd .., cd ~, cd path. Also using git and it's commands to control my repo and the branches.



```
ties  Terminal Sep 22 17:00
carlos@carlos-virtual-machine: ~/Carlos_CepedaXV6
carlos@carlos-virtual-machine: ~
create mode 100644 xv6-riscv-labs.git/hooks/pre-receive.sample
create mode 100644 xv6-riscv-labs.git/hooks/prepare-commit-msg.sample
create mode 100644 xv6-riscv-labs.git/hooks/push-to-checkout.sample
create mode 100644 xv6-riscv-labs.git/hooks/update.sample
create mode 100644 xv6-riscv-labs.git/info/exclude
create mode 100644 xv6-riscv-labs.git/objects/pack/pack-4f892e694b55b3cc29aa39e
bc57113be3ab3d0ff.idx
create mode 100644 xv6-riscv-labs.git/objects/pack/pack-4f892e694b55b3cc29aa39e
bc57113be3ab3d0ff.pack
create mode 100644 xv6-riscv-labs.git/packed-refs
carlos@carlos-virtual-machine:~/Carlos_CepedaXV6$ git checkout HW1
Already on 'HW1'
Your branch is up to date with 'origin/HW1'.
carlos@carlos-virtual-machine:~/Carlos_CepedaXV6$ ls
fs.img  kernel  LICENSE  Makefile  mkfs  README  user  xv6-riscv-labs.git
carlos@carlos-virtual-machine:~/Carlos_CepedaXV6$ cd user
carlos@carlos-virtual-machine:~/Carlos_CepedaXV6/user$ ls
_cat      grind.asm  ln.asm    ps.o      sh.o      usertests.c
cat.asm   grind.c    ln.c      ps.sym    sh.sym    usertests.d
cat.c     grind.d    ln.d      _ptest    _sleep    usertests.o
cat.d     grind.o    ln.o      ptest.asm sleep.asm  usertests.sym
cat.o     grind.sym  ln.sym    ptest.c   sleep.c   usys.d
cat.sym   _init     _ls       ptest.d   sleep.d   usys.o
_echo     init.asm  ls.asm    ptest.o   sleep.o   usys.pl
```



A terminal window titled 'carlos@carlos-virtual-machine: ~' showing the boot process of the xv6 kernel. The output indicates that the kernel is booting, followed by the starting of hart 1 and hart 2. The 'init' process starts a shell ('sh'). The user then runs the 'ls' command, which displays a directory listing of files and processes, including their permissions, owner, group, size, and name.

```
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
```

Task 2. Implement the uptime utility.

Once everything was setup creating the file was not hard. This gave me a a good understanding of how we're going to be conducting the homework and projects for the class. For the most part the only thing I had an issue with was understanding the instructions. But with the help of other students, we realized what needed to be done. The coding part was straightforward.

```

Sep 22 16:36
carlos@carlos-virtual-machine: ~/Carlos_CepedaXV6
carlos@carlos-virtual-machine: ~
carlos@carlos-virtual-machine: ~/Carlos...
ser/uptime.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_uptime > user/uptime.asm
riscv64-linux-gnu-objdump -t user/_uptime | sed '1,/SYMBOL TABLE/d; s/ .* / /; /
^$/d' > user/uptime.sym
mkfs/mkfs fs.img README user/_cat user/_echo user/_forktest user/_grep user/_ini
t 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/_ptest user/_uptime
nmeta 46 (boot, super, log blocks 30 inode blocks 13, bitmap blocks 1) blocks 95
4 total 1000
ballocc: first 706 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-devi
ce,drive=x0,bus=virtio-mmio-bus.0

xv6 kernel is booting

hart 2 starting
hart 1 starting
init: starting sh
$ uptime
up 40 clock ticks
$
```