<u>Name:</u> Ali Rida Awad

<u>ID:</u> 202401137

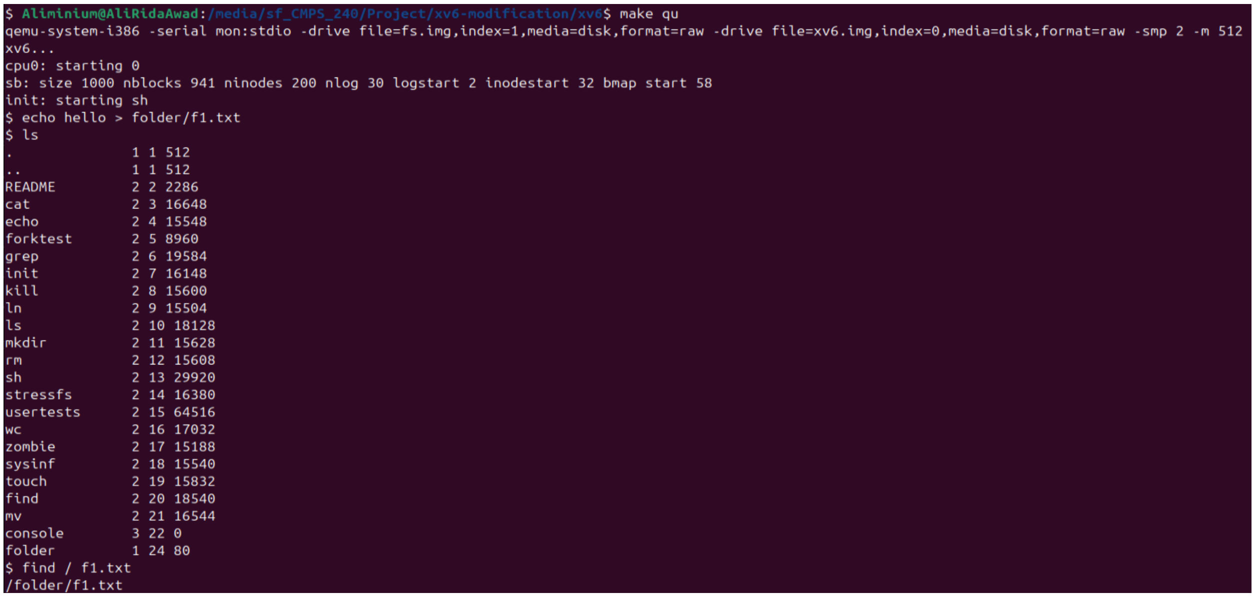
<u>Subject:</u> CMPS 240 Project Sprint 2

Implemented features:

1. “find” command, takes as parameters a directory and the file name, and searches for the file in the taken directory. (Added file “find.c”, and added it to the makefile)
2. “mv” command, takes as parameters two file names or paths to files, moves the first file into the second file. (Added file “mv.c”, and added it to the makefile)
3. “touch” command, which takes a file name (or path to file) as a parameter and creates a file in the given path. (Added file “touch.c”, and added it to the makefile)

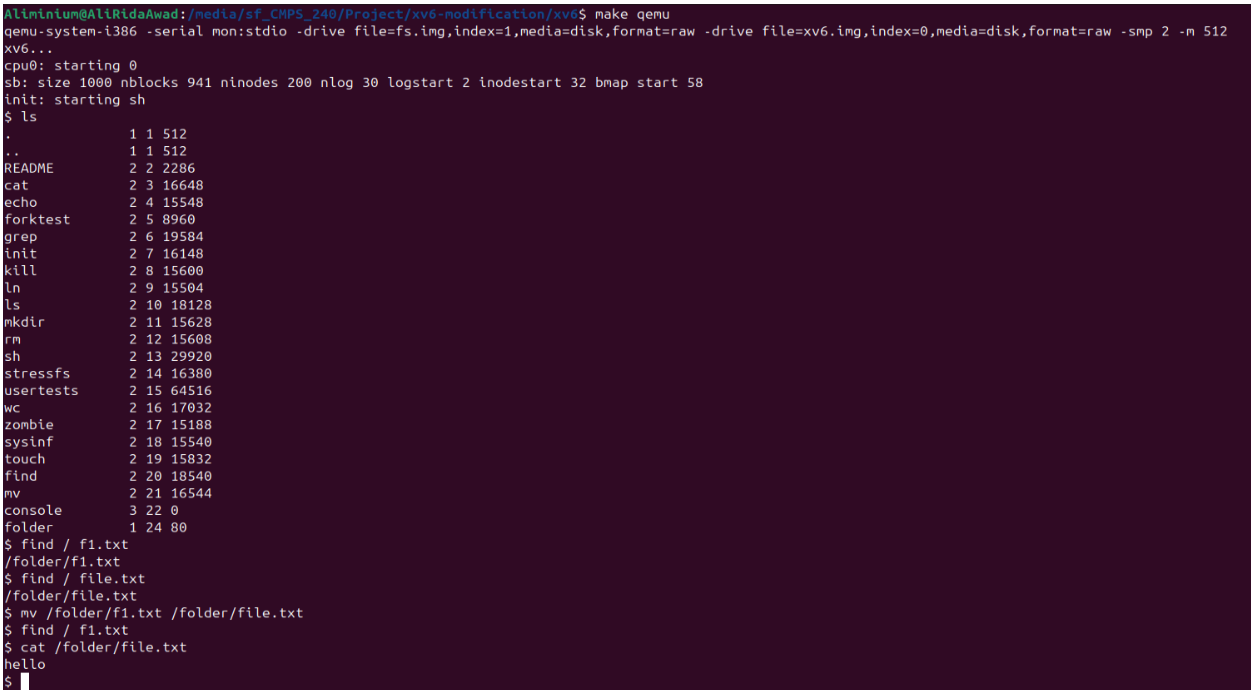
Implementation:

1. “find”: This feature was relatively long but simple. It takes the given path parameter and recursively searches all child directories for the file with the same file name as passed in its parameters. After finding the file, it prints the directory containing it to the user.



```
$ Aliminium@AliRidaAwad:/media/sf_CMPS_240/Project/xv6-modification/xv6$ make qu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ echo hello > folder/f1.txt
$ ls
.              1 1 512
..             1 1 512
README         2 2 2286
cat            2 3 16648
echo           2 4 15548
forktest       2 5 8960
grep           2 6 19584
init           2 7 16148
kill           2 8 15600
ln             2 9 15504
ls             2 10 18128
mkdir          2 11 15628
rm             2 12 15608
sh             2 13 29920
stressfs       2 14 16380
usertests      2 15 64516
wc             2 16 17032
zombie         2 17 15188
sysinf         2 18 15540
touch          2 19 15832
find           2 20 18540
mv             2 21 16544
console        3 22 0
folder         1 24 80
$ find / f1.txt
/folder/f1.txt
```

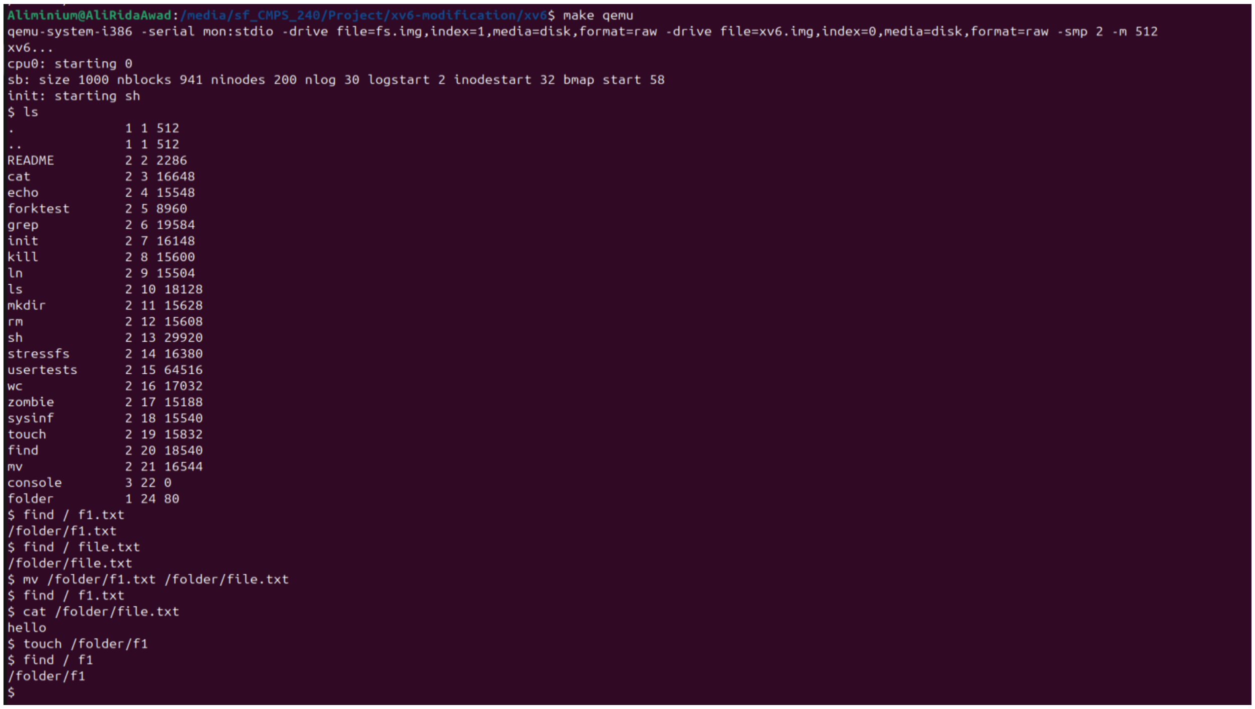
2. “mv”: This feature was implemented by using XV6’s open and close commands with a 512-character buffer. It opens the source and destination files, reads 512 characters into the buffer from the source file, writes the buffer to the destination file, and repeats until the source file is empty. After which, it deletes the source file through the “unlink” user call.



```
Aliminium@AliRidaAwad:/media/sf_CMPS_240/Project/xv6-modification/xv6$ make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ ls
.              1 1 512
..             1 1 512
README         2 2 2286
cat            2 3 16648
echo           2 4 15548
forktest       2 5 8960
grep           2 6 19584
init           2 7 16148
kill           2 8 15600
ln             2 9 15504
ls             2 10 18128
mkdir          2 11 15628
rm             2 12 15608
sh             2 13 29920
stressfs       2 14 16380
usertests      2 15 64516
wc             2 16 17032
zombie         2 17 15188
sysinf         2 18 15540
touch          2 19 15832
find           2 20 18540
mv             2 21 16544
console        3 22 0
folder         1 24 80
$ find / f1.txt
/folder/f1.txt
$ find / file.txt
/folder/file.txt
$ mv /folder/f1.txt /folder/file.txt
$ find / f1.txt
$ cat /folder/file.txt
hello
$ 
```

(Keep in mind that in the picture above, the f1.txt file contains hello as it was created with the previous screenshot's "echo hello > folder/f1.txt" command)

3. "touch": Its implementation is extremely simple. It opens the passed parameter (file or path to file) as a file using the "open" user call, then closes the file if no errors occurred.



```
Aliminium@AliRidaAwad:/media/sf_CMPS_240/Project/xv6-modification/xv6$ make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ ls
.              1 1 512
..             1 1 512
README         2 2 2286
cat            2 3 16648
echo           2 4 15548
forktest       2 5 8960
grep           2 6 19584
init           2 7 16148
kill           2 8 15600
ln             2 9 15504
ls             2 10 18128
mkdir          2 11 15628
rm             2 12 15608
sh             2 13 29920
stressfs       2 14 16380
usertests      2 15 64516
wc             2 16 17032
zombie         2 17 15188
sysinf         2 18 15540
touch          2 19 15832
find           2 20 18540
mv             2 21 16544
console        3 22 0
folder         1 24 80
$ find / f1.txt
/folder/f1.txt
$ find / file.txt
/folder/file.txt
$ mv /folder/f1.txt /folder/file.txt
$ find / f1.txt
$ cat /folder/file.txt
hello
$ touch /folder/f1
$ find / f1
/folder/f1
$
```