# Project 4 Test Report

Alexander DuPree

March 2, 2020

# Introduction

The following test report documents the tests performed for project four. The test cases and strategies closely follow the project four rubric.

Each section contains test cases related to the sections topic. Each test case will describe the name of the test, the expected result, actual result, as well as a discussion and indication of the Pass/Fail status. The actual result will be provided in the form of a screen shot of the console.

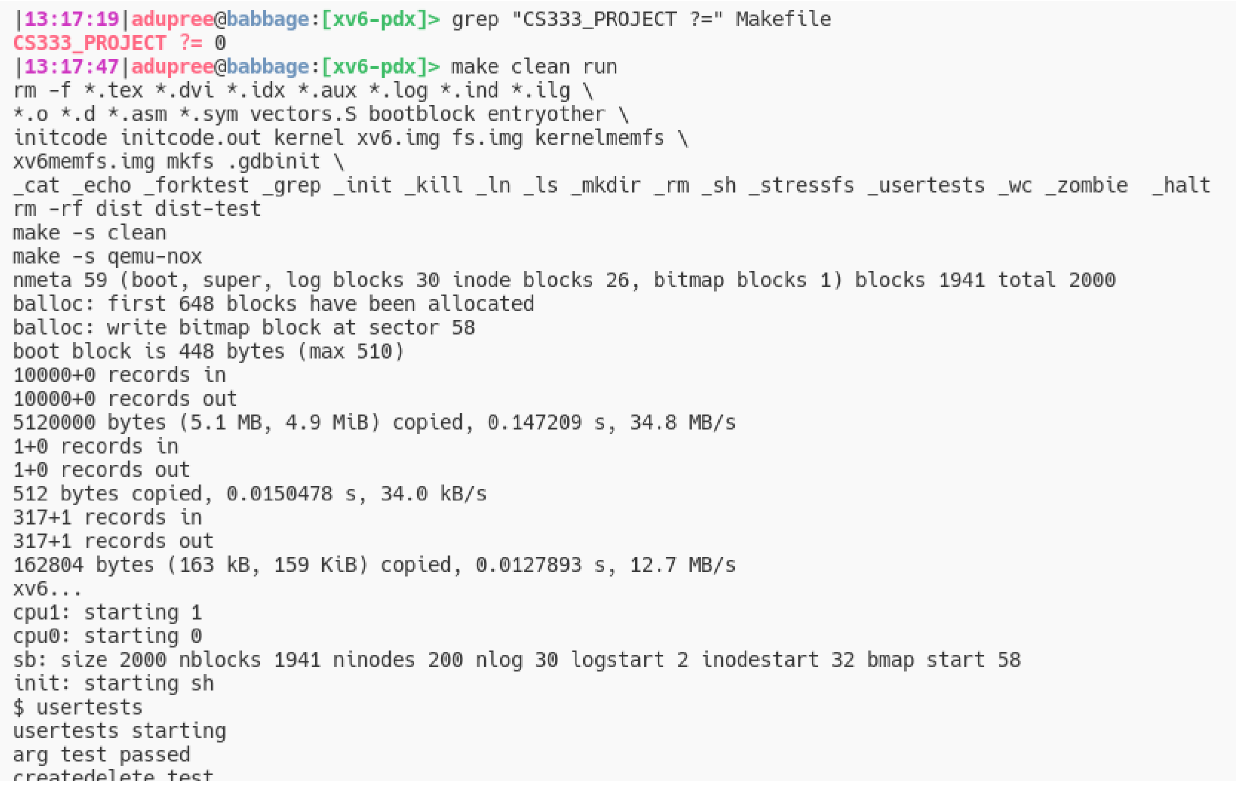
# Compilation

This section presents all tests related to compiling the xv6 kernel. Test cases follow closely those outlined in the rubric.

**Test Case:** *With CS333_PROJECT set to 0 in the Makefile*
**Assertions:**

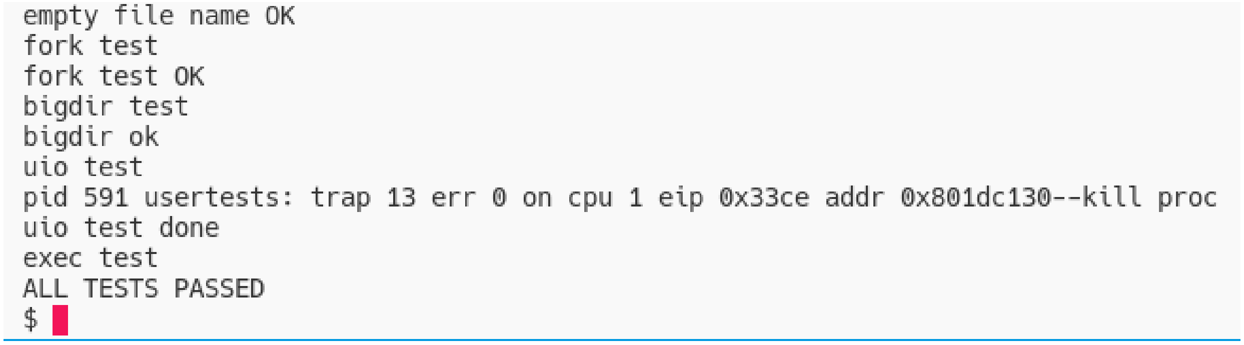
1. Code correctly compiles
2. Kernel successfully boots
3. `usertests` run to completion with all tests passed

**Status:** PASS



```
|13:17:19|adupree@babbage:[xv6-pdx]> grep "CS333_PROJECT ?=" Makefile
CS333_PROJECT ?= 0
|13:17:47|adupree@babbage:[xv6-pdx]> make clean run
rm -f *.tex *.dvi *.idx *.aux *.log *.ind *.ilg \
*.o *.d *.asm *.sym vectors.S bootblock entryother \
initcode initcode.out kernel xv6.img fs.img kernelmemfs \
xv6memfs.img mkfs .gdbinit \
_cat _echo _forktest _grep _init _kill _ln _ls _mkdir _rm _sh _stressfs _usertests _wc _zombie  _halt
rm -rf dist dist-test
make -s clean
make -s qemu-nox
nmeta 59 (boot, super, log blocks 30 inode blocks 26, bitmap blocks 1) blocks 1941 total 2000
balloc: first 648 blocks have been allocated
balloc: write bitmap block at sector 58
boot block is 448 bytes (max 510)
10000+0 records in
10000+0 records out
5120000 bytes (5.1 MB, 4.9 MiB) copied, 0.147209 s, 34.8 MB/s
1+0 records in
1+0 records out
512 bytes copied, 0.0150478 s, 34.0 kB/s
317+1 records in
317+1 records out
162804 bytes (163 kB, 159 KiB) copied, 0.0127893 s, 12.7 MB/s
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 2000 nblocks 1941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ usertests
usertests starting
arg test passed
createdelete test
```

Figure 1: Compilation and boot with CS333_PROJECT set to 0 and execution of `usertests`



```
empty file name OK
fork test
fork test OK
bigdir test
bigdir ok
uio test
pid 591 usertests: trap 13 err 0 on cpu 1 eip 0x33ce addr 0x801dc130--kill proc
uio test done
exec test
ALL TESTS PASSED
$ 
```

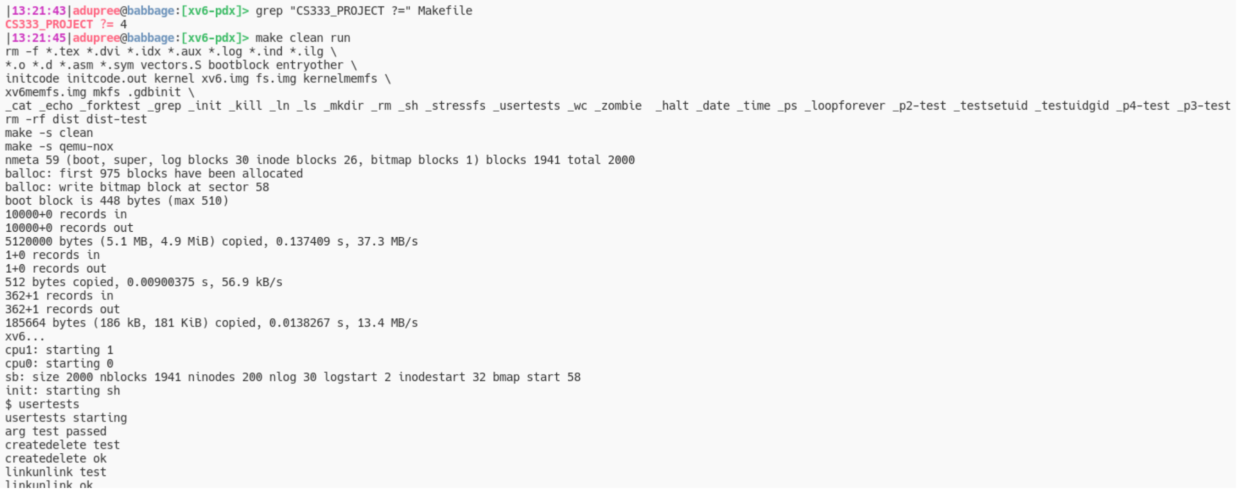
Figure 2: Completion of `usertests` with output elided

The command `grep "CS333_PROJECT ?=" Makefile` shows that the CS333_PROJECT macro is set to 0. The following command `make clean run` demonstrates that the code correctly compiles and the kernel successfully boots. Furthermore, the commands were executed within seconds of each other, indicating that tampering is not a possibility. Lastly, we can see that the execution of `usertests` is initiated in the same session, and Figure 2 shows that the tests run to completion and all tests pass.

**Test Case:** *With CS333_PROJECT set to 4 in the Makefile*
**Assertions:**

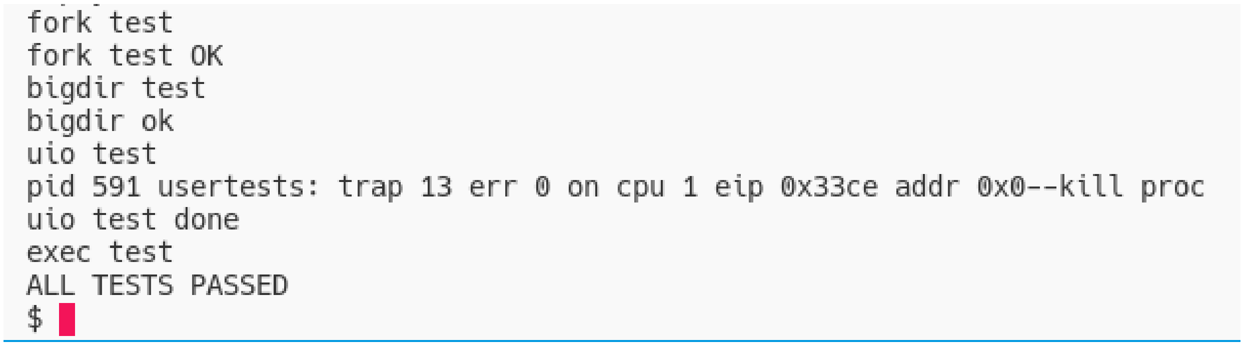
1. Code correctly compiles
2. Kernel successfully boots
3. `usertests` run to completion with all tests passed

**Status:** **PASS**



```
|13:21:43|adupree@babbage:[xv6-pdx]> grep "CS333_PROJECT ?=" Makefile
CS333_PROJECT ?= 4
|13:21:45|adupree@babbage:[xv6-pdx]> make clean run
rm -f *.tex *.dvi *.idx *.aux *.log *.ind *.ilg \
*.o *.d *.asm *.sym vectors.S bootblock entryother \
initcode initcode.out kernel xv6.img fs.img kernelmemfs \
xv6memfs.img mkfs .gdbinit \
_cat _echo _forktest _grep _init _kill _ln _ls _mkdir _rm _sh _stressfs _usertests _wc _zombie  _halt _date _time _ps _loopforever _p2-test _testsetuid _testuidgid _p4-test _p3-test
rm -rf dist dist-test
make -s clean
make -s qemu-nox
nmeta 59 (boot, super, log blocks 30 inode blocks 26, bitmap blocks 1) blocks 1941 total 2000
balloc: first 975 blocks have been allocated
balloc: write bitmap block at sector 58
boot block is 448 bytes (max 510)
10000+0 records in
10000+0 records out
5120000 bytes (5.1 MB, 4.9 MiB) copied, 0.137409 s, 37.3 MB/s
1+0 records in
1+0 records out
512 bytes copied, 0.00900375 s, 56.9 kB/s
362+1 records in
362+1 records out
185664 bytes (186 kB, 181 KiB) copied, 0.0138267 s, 13.4 MB/s
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 2000 nblocks 1941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ usertests
usertests starting
arg test passed
createdelete test
createdelete ok
linkunlink test
linkunlink ok
```

Figure 3: Compilation and boot with CS333_PROJECT set to 4 and execution of `usertests`.



```
fork test
fork test OK
bigdir test
bigdir ok
uio test
pid 591 usertests: trap 13 err 0 on cpu 1 eip 0x33ce addr 0x0--kill proc
uio test done
exec test
ALL TESTS PASSED
$ 
```

Figure 4: Completion of `usertests` with output elided, CS333_P4 is defined

The command `grep "CS333_PROJECT ?=" Makefile` shows that the CS333_PROJECT macro is indeed set to 4. The following command `make clean run` demonstrates that the code correctly compiles and the kernel successfully boots. Furthermore, the commands were executed within seconds of each other, indicating that tampering is not a possibility. Lastly, we can see that the execution of `usertests` is initiated in the same session and Figure 4 demonstrates that the tests run to completion and all tests pass.