Compilation Test

This section will show that my project one code compiles with the project flag set to “1” and also set to “0”.
Subtest 1: Compile with CS333 PROJECT set to 0. Since the listing is so long, this will require two screen shots. In the first screen shot, the current date and time is displayed as well as the value for CS333 PROJECT, verifying that it is set to 0. In the second screen show, the same information is displayed. This is used to show that the two screen shots are from the same compilation sequence. The expected outcome is that the compilation step will correctly compile with the project flag set to “0”.

```
jiacheng@babbage:~/foo/xv6-pdx$ date
Sun Apr 21 15:54:21 PDT 2019
jiacheng@babbage:~/foo/xv6-pdx$ grep "CS333_PROJECT ?=" Makefile
CS333_PROJECT ?= 0
jiacheng@babbage:~/foo/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 qemu-nox
make[1]: Entering directory '/u/jiacheng/foo/xv6-pdx'
gcc -Werror -Wall -DPDX_XV6 -o mkfs mkfs.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O1 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -DPDX_XV6 -fno-pie -no-pie   -c -o ulib.o ulib.
```

Compilation with CS333 PROJECT set to 0

```
165132 bytes (165 kB, 161 KiB) copied, 0.0137786 s, 12.0 MB/s
qemu-system-i386 -nographic -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...
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
$ halt
Shutting down ...
make[1]: Leaving directory '/u/jiacheng/foo/xv6-pdx'
jiacheng@babbage:~/foo/xv6-pdx$ grep "CS333_PROJECT ?=" Makefile
CS333_PROJECT ?= 0
jiacheng@babbage:~/foo/xv6-pdx$ date
Sun Apr 21 15:55:03 PDT 2019
```

Compilation with CS333 PROJECT set to 0

```
$
1       init    sleep    8010387f 80103989 80104d29 8010415a 80104f3e 80104e5a
2       sh      sleep    80103843 801002b1 80101857 80100e4e 80104432 8010415a 80104f3e 80104e5a
ps
exec ps failed
$ halt
Shutting down ...
make[1]: Leaving directory '/u/jiacheng/foo/xv6-pdx'
jiacheng@babbage:~/foo/xv6-pdx$ grep "CS333_PROJECT ?=" Makefile
CS333_PROJECT ?= 0
```

Compilation with CS333 PROJECT set to 0

The date in the first and second figures show about 12 seconds of elapsed time. This shows that the two date commands occurred close in time. The grep commands before and after the compilation show that the project flag in the Makefile is set to “0”. The date commands are executed close in time, the project flag shows the same value before and after the compilation, and the compilation shows no errors. This leads to the conclusion that the project code correctly compiles with the project flag turned off.
This subtest PASSES.

Subtest 2: Boot compile with CS333 PROJECT set to 2. Since the listing is so long, this will require two screen shots. In the first screen shot, the current date and time is displayed as well as the value for CS333 PROJECT, verifying that it is set to 2. In the second screen show, the same information is displayed. This is used to show that the two screen shots are from the same compilation sequence. The expected outcome is that the compilation step will correctly compile with the project flag set to "2".

```
jiacheng@babbage:~/foo/xv6-pdx$ date
Sun Apr 21 23:03:20 PDT 2019
jiacheng@babbage:~/foo/xv6-pdx$ grep "CS333_PROJECT ?=" Makefile
CS333_PROJECT ?= 2
jiacheng@babbage:~/foo/xv6-pdx$ make
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O1 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -DPDX_XV6 -DCS333_P1 -DUSE_BUILTINS -DCS333_P2
-fno-pie -no-pie -fno-pic -O -nostdinc -I. -c bootmain.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O1 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -DPDX_XV6 -DCS333_P1 -DUSE_BUILTINS -DCS333_P2
-fno-pie -no-pie -fno-pic -nostdinc -I. -c bootasm.S
ld -m    elf_i386 -N -e start -Ttext 0x7C00 -o bootblock.o bootasm.o bootmain.o
objdump -S bootblock.o > bootblock.asm
objcopy -S -O binary -j .text bootblock.o bootblock
./sign.pl bootblock
```

Compilation with CS333 PROJECT set to 2

```
----- - ------- ---
5120000 bytes (5.1 MB, 4.9 MiB) copied, 0.158583 s, 32.3 MB/s
dd if=bootblock of=xv6.img conv=notrunc
1+0 records in
1+0 records out
512 bytes copied, 0.0100999 s, 50.7 kB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
337+1 records in
337+1 records out
172860 bytes (173 kB, 169 KiB) copied, 0.0144119 s, 12.0 MB/s
jiacheng@babbage:~/foo/xv6-pdx$ grep "CS333_PROJECT ?=" Makefile
CS333_PROJECT ?= 2
jiacheng@babbage:~/foo/xv6-pdx$ date
Sun Apr 21 23:05:57 PDT 2019
```

Compilation with CS333 PROJECT set to 2

The date in the first and second figures show about 10 seconds of elapsed time. This shows that the two date commands occurred close in time. The grep commands before and after the compilation show that the project flag in the Makefile is set to "2". The date commands are CS333 Project 2 Test Report Your Name Here executed close in time, the project flag shows the same value before and after the compilation, and the compilation shows no errors. This leads to the conclusion that the project code correctly compiles with the project flag turned off.
This subtest PASSES.
Since each subtest passes and the subtests fully test the objectives, this test PASSES.

Usertests and forktest run with CS333 P1 flag turned on

Subtest 3: I tested that xv6 correctly compiles and runs with the CS333 P2 flag enabled. I set the CS333 PROJECT value in the Makefile to 2, compiled and booted xv6 using make qemu-nox, and then ran usertests. As mentioned above, this is an acceptable test for both usertests and forktest since forktest is run as part of usertests. It is expected that all tests from usertests will pass.

```
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
$ date
Mon Apr 22 06:13:20 UTC 2019
$ forktest
fork test
fork test OK
$ usertests
usertests starting
arg test passed
createdelete test
createdelete ok
```

forktest with CS333 P2 enabled

```
unlinkread test
unlinkread ok
dir vs file
dir vs file OK
empty file name
empty file name OK
fork test
fork test OK
bigdir test
bigdir ok
uio test
pid 654 usertests: trap 13 err 0 on cpu 1 eip 0x340d addr 0x801dc130--kill proc
uio test done
exec test
ALL TESTS PASSED
$ date
Mon Apr 22 06:15:19 UTC 2019
```

Usertests with CS333 P2 enabled

From both figures, we can see that all usertests have passed. Further, since forktest is run as a part of usertests, we know that forktest has passed. This test PASSES.

UID/GID test

Subtest 4: CS333_PROJECT flag set to 2. Run UID/GID test. The excepted result is come out new PID = 4 process testuidgid and the set is 100. Then, test set GID will get new process with PID = 5 and the set is 111.

```
$ testuidgid
Current UID is: 0
Setting UID to 100
Current UID is: 100

PID     Name    UID     GID     PPID    Elapsed CPU     State   Size     PCs
1       init    0       0       0       33.588  0.7     sleep   12288    801038f1 80103a01 801050a5 801044d3 801053ee 8010530a
2       sh      0       0       0       33.547  0.9     sleep   16384    801038f1 80103a01 801050a5 801044d3 801053ee 8010530a
4       testuidgid      100     0       0       3.37    0.0     run     12288
Current GID is: 0
Setting GID to 200
Current GID is: 200

PID     Name    UID     GID     PPID    Elapsed CPU     State   Size     PCs
1       init    0       0       0       37.119  0.7     sleep   12288    801038f1 80103a01 801050a5 801044d3 801053ee 8010530a
2       sh      0       0       0       37.78   0.9     sleep   16384    801038f1 80103a01 801050a5 801044d3 801053ee 8010530a
4       testuidgid      100     200     0       6.568   0.0     sleep   12288    8010390f 8010516f 801044d3 801053ee 8010530a
My parent process is: 2
Setting UID to 111 and GID to 111 before fork(). Value should be inherited
Before fork(), UID = 111, GID = 111
Child: UID is: 111, GID is: 111

PID     Name    UID     GID     PPID    Elapsed CPU     State   Size     PCs
1       init    0       0       0       42.709  0.7     sleep   12288    801038f1 80103a01 801050a5 801044d3 801053ee 8010530a
2       sh      0       0       0       42.668  0.9     sleep   16384    801038f1 80103a01 801050a5 801044d3 801053ee 8010530a
4       testuidgid      111     111     0       12.158  0.0     sleep   12288    8010390f 8010516f 801044d3 801053ee 8010530a
5       testuidgid      111     111     0       2.59    0.0     sleep   12288    8010390f 8010516f 801044d3 801053ee 8010530a
Setting UID to 32800. This test should FAIL
SUCCESS! The setuid sytem call indicated failure
Setting GID to 32800. This test should FAIL
SUCCESS! The setgid sytem call indicated failure
Setting UID to -1. This test should FAIL
SUCCESS! The setuid sytem call indicated failure
Done!
$ zombie!
```

The picture above show the result of testuidgid test which is the excepted result.
UID/GID test PASSES.

Ps and control-p command test

Subtest 5: CS333_PROJECT flag set to 2. Run ps and control-p test. The control-p show the PID, Name,UID,GID,PPID, Elapsed time, CPU, State, Size and PCs. The ps show the PID, Name,UID,GID,PPID, Elapsed time, CPU, State and Size.

```
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
$
PID     Name    UID     GID     PPID    Elapsed CPU     State   Size    PCs
1       init    0       0       0       10.897  0.7     sleep   12288   801038f1 80103a01 801050a5 801044d3 801053ee 8010530a
2       sh      0       0       0       10.856  0.3     sleep   16384   801038b5 801002b1 80101858 80100e4f 801047ae 801044d3 801053ee 8010530a
ps

PID     Name    UID     GID     PPID    Elapsed CPU     State   Size
1       init    0       0       1       12.921  0.7     sleep   12288
2       sh      0       0       1       12.880  0.6     sleep   16384
3       ps      0       0       2       0.21    0.2     run     45056
```

The picture above show the result of ps and control-p test which is the excepted result.

ps and control-p command test PASSES.

Time command test

Subtest 6: By using p2-test with CS333_PROJECT flag set to 2. This test uses the built-in p2-test function to test the time command runs correctly without any observational errors.

```
----------
Running Time Test
----------
You will need to verify these tests passed

time
time ran in 0.000 seconds

time abc
abc ran in 0.007 seconds

time date
Mon Apr 22 05:06:36 UTC 2019
date ran in 0.022 seconds

time time echo "abc"
"abc"
echo ran in 0.018 seconds
time ran in 0.037 seconds
```

The picture above show the result of command test which is the excepted result.

Time command PASSES.

Getprocs test with p2-test

Subtest 7: CS333_PROJECT flag set to 2. Run p2-test, it shows getprocs test

```
----------
Running UID / GID Tests
----------
** All tests passed! **

----------
Running UID / GID Inheritance Test
----------
** Test Passed! **

----------
Running PPID Test
----------
** Test passed! **
```

UID/GID test in p2-test

```
----------
Running GetProcs Test
----------
Filling the proc[] array with dummy processes
ERROR
----------
```

Getprocs test in p2-test

The pictures above show the result of getprocs test which is the excepted result.
This test is FAIL.