Operating Systems Design Skilling Assignment
19CS2106A

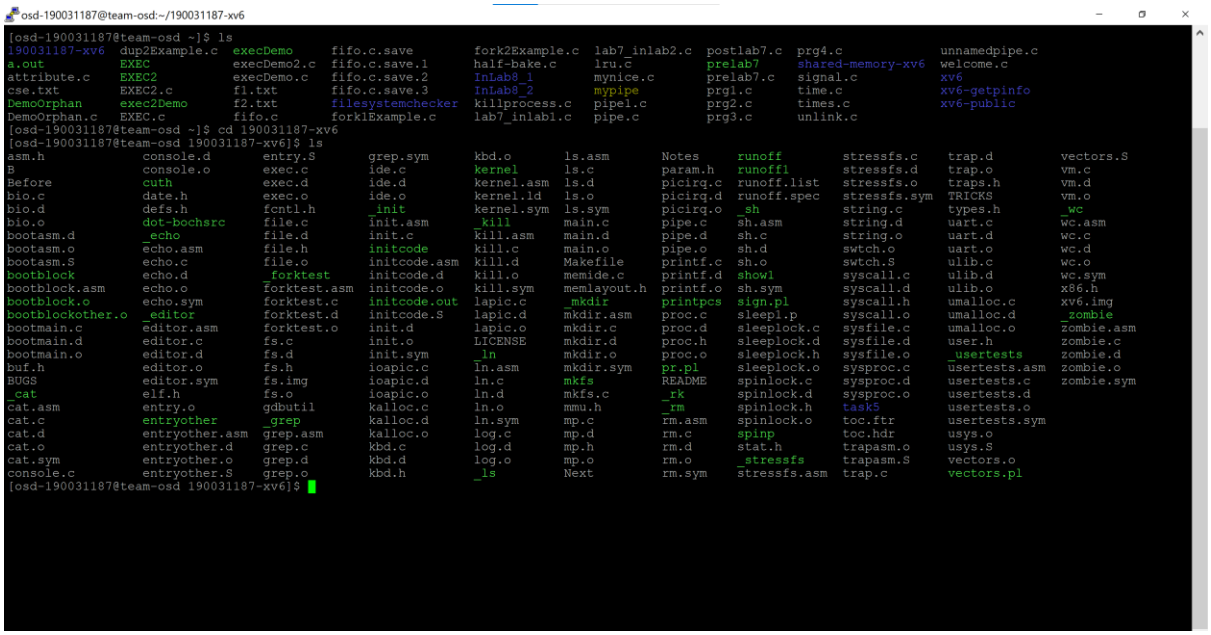
Project no. 8

Tasks to be performed:

1) Improved Shell
2) Improved Editor
3) Add System call ls – to list files in xv6
4) Enchantment - Porting xv6 with POSIX compliance + VFS + ulibc + ACPI
5) Add System Call test for Mirrored Raid Feature

Initially, we clone into a new xv6 for specific purpose of this project by using command
→$git clone git://github.com/mit-pdos/xv6-public.git 190031187-xv6

Step into the specific xv6 – by using command cd 190031187-xv6



Task 1: Improvised Shell for XV6.

About Shell:

The shell is a program that takes commands from the keyboard and gives them to the operating system to perform. A shell is accessed by a terminal which runs it.

As, there is a shell code in the newly created xv6 move the old shell code and place a new improvised shell code by using the following commands.

→mv sh.c oldsh.c
→nano sh.c

Code in Putty Server: The new sh.c will be having the improvised shell code.

Execution and Result of Task 1:

→ make qemu-nox

osd-190031187@team-osd:~/190031187-xv6

```
SeaBIOS (version 1.11.0-2.el7)


iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF94780+1FED4780 C980



Booting from Hard Disk..xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
190031187$ ls
.              1 1 512
..             1 1 512
README         2 2 2286
cat            2 3 14480
echo           2 4 13336
forktest       2 5 8160
grep           2 6 16016
init           2 7 14224
kill           2 8 13368
ln             2 9 13308
ls             2 10 16168
mkdir          2 11 13400
rm             2 12 13376
sh             2 13 31064
stressfs       2 14 14324
usertests      2 15 67224
wc             2 16 15144
zombie         2 17 13036
editor         2 18 26800
console        3 19 0
OSD            1 20 48
f1.txt         2 21 16
190031187$ cd OSD
190031187$ pwd
/OSD
190031187$ ls
.              1 20 48
..             1 1 512
f2.txt         2 22 13
190031187$
```

Observation and Analysis:

In Old shell pwd and ls was not working inside a directory. Here we observe, that an improvised shell is being used and was checked if properly working or not by executing one of the commands from list above.

→mkdir OSD
→cd OSD
→pwd
→ls

Task 2: Improvised Editor in XV6.

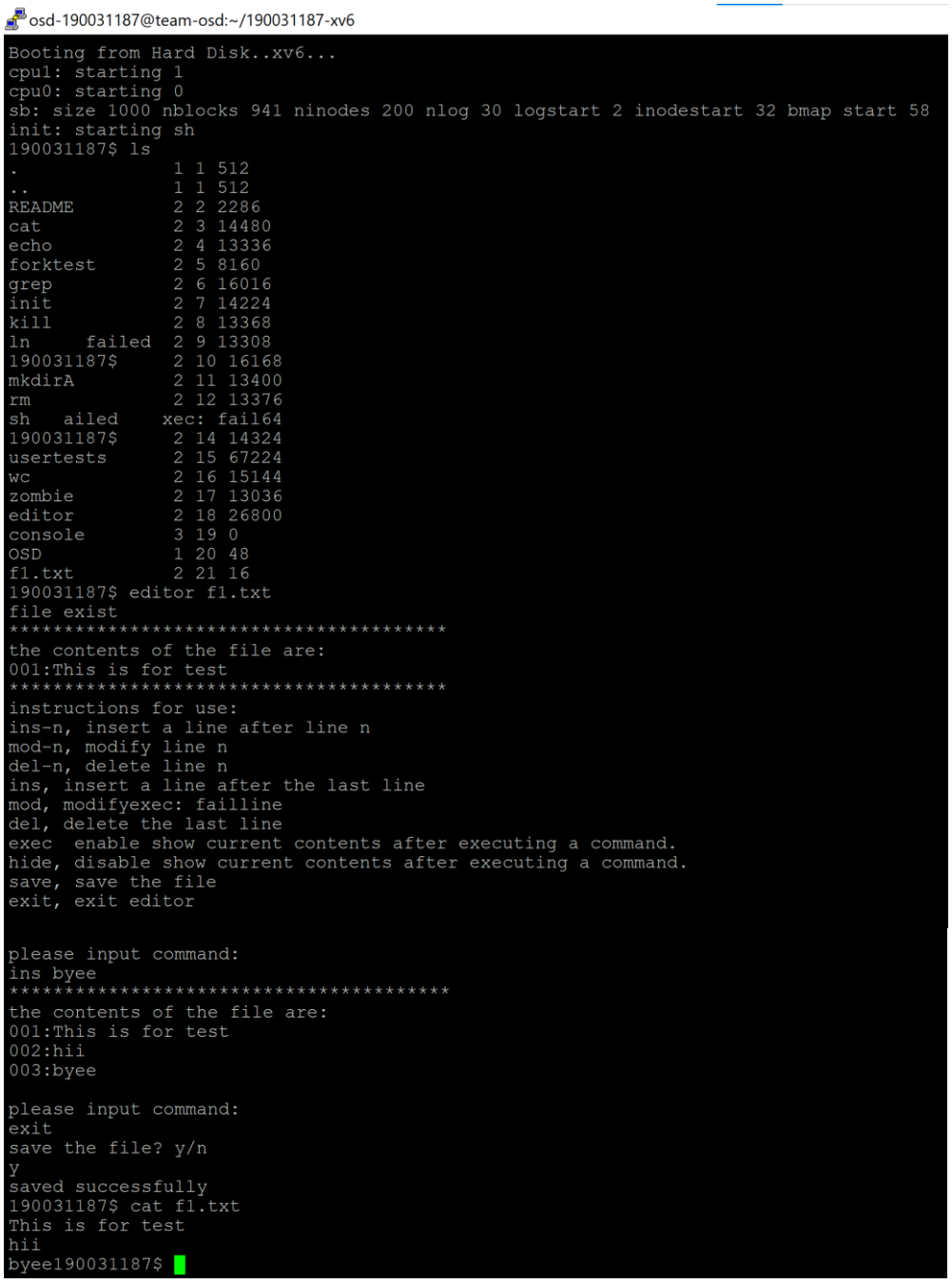
About Editor:

The vi editor is the most popular and commonly used Unix text editor. It is usually available in all Linux Distributions. It works in two modes, Command and insert. Command mode takes the user commands, and the Insert mode is for editing text.

Here we create an improvised Editor for XV6 where it can insert a text in any particular line asked by user and at the same time can delete a particular text from any given line. It not only has Insert and Delete functions but also can show and hide contexts of text and file.

Code in Putty Server: editor.c

Execution and Result of Task 2:



Observation and Analysis:

Here a file f1.txt has be created initially by ‘cat>f1.txt’ later was modified by using the editor command. After execution the editor also asks whether to save the file or not and exits the editor by command exit. We can also try the ins-n command which takes a parameter called line from user to insert or delete in a particular line.

Task 3: Adding System Call ls

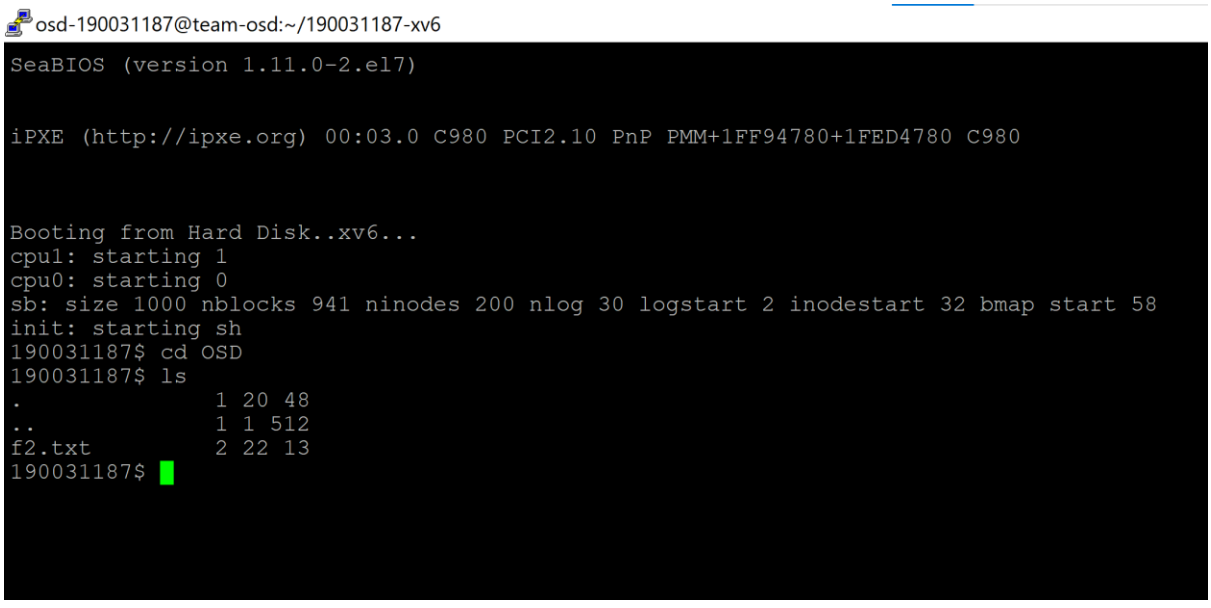
About ls:

Lists the names of files in a particular Unix directory. If you type the ls command with no parameters or qualifiers, the command displays the files listed in your current working directory. There are many other different ways of listing out the files for example:

ls -l: list with long format - show permissions
ls -s: list file size
ls -S: sort by file size
ls -a: list all files including hidden files starting with ‘.’

Code in Putty Server: ls.c

Execution and Result of Task 3:



Observation and Analysis:

Here we observe the ls command working by creating a folder/directory by using ‘mkdir’ command and creating a sample.txt file in the directory and use ls to list the text file.

Task 4: Enchantment - Porting xv6 with POSIX compliance + VFS + ulibc + ACPI

Some definitions:

POSIX - The Portable Operating System Interface (POSIX) is an IEEE standard that helps compatibility and portability between operating systems. Theoretically, POSIX compliant source code should be seamlessly portable. In the real world, application transition often runs into system specific issues.

VFS - The Virtual File System (also known as the Virtual Filesystem Switch) is the software layer in the kernel that provides the filesystem interface to user space programs. It also provides an abstraction within the kernel which allows different filesystem implementations to coexist.

Ulibc - It is a small C standard library intended for Linux kernel-based operating systems for embedded systems and mobile devices.

ACPI - ACPI, known as a Hardware Abstraction Layer (HAL) in embedded computing, is an abstraction layer between the operating system, platform firmware and hardware. This allows the OS and the platform to evolve independently. The core of the Linux ACPI implementation comes from ACPICA (ACPI Component Architecture).

These are basically enchantments (special features) to help in improvising a better a xv6.

Task 5: Add System Call test for Mirrored Raid Feature.

About Mirrored Raid:

RAID Mirroring means an exact clone (or mirror) of the same data writing to two drives. A minimum two number of disks are more required in an array to create RAID1 and it's useful only, when read performance or reliability is more precise than the data storage capacity.

Here we perform two tests for Mirrored Raid Feature.

Code in Putty Server: tester.c and write2.c

Execution and Result of Task 5:

osd-190031187@team-osd:~/190031187-xv6/task5

```
Booting from Hard Disk..xv6...
lapicinit: 1 0xfee00000
cpu1: starting
cpu0: starting
init: starting sh
190031187$ tester
mirrored fs test
write done
read text = xxxxxxxxxx 190031187$
190031187$ write2
Writing A
Writing B
Writing C
Writing D
Writing E
Writing F
Writing G
Writing H
Writing I
Writing J
Writing K
Writing L
Writing M
Writing N
Writing O
Writing P
Writing Q
Writing R
Writing S
Writing T
Writing U
Writing V
Writing W
Writing X
Writing Y
Writing Z
Writing [
Writing \
Writing ]
Writing ^
Writing _
Writing `
Writing a
Writing b
Writing c
```

```
Writing d
Writing e
Writing f
Writing g
Writing h
Writing i
Writing j
Writing k
Writing l
Writing m
Writing n
Writing o
Writing p
Writing q
Writing r
Writing s
Writing t
Writing u
Writing v
Writing w
Writing x
Writing y
Writing z
Writing {
Writing |
Writing }
Writing ~
Writing
Writing ▒
Writing ▒
Writing ▒
Writing ▒
Writing ▒
Writing ▒
Writing ▒
Reading A
Reading B
Reading C
Reading D
Reading E
Reading F
Reading G
Reading H
Reading I
Reading J
Reading K
```

```
Writing L
Writing M
Writing N
Writing O
Writing P
Writing Q
Writing R
Writing S
Writing T
Writing U
Writing V
Writing W
Writing X
Writing Y
Writing Z
Writing [
Writing \
Writing ]
Writing ^
Writing _
Writing `
Writing a
Writing b
Writing c
Writing d
Writing e
Writing f
Writing g
Writing h
Writing i
Writing j
Writing k
Writing l
Writing m
Writing n
Writing o
Writing p
Writing q
Writing r
Writing s
Writing t
Writing u
Writing v
Writing w
Writing x
Writing y
```

```
Writing z
Writing {
Writing |
Writing }
Writing ~
Writing
Writing ▒
Writing ▒
Writing ▒
Writing ▒
Writing ▒
Writing ▒
Writing ▒
Reading A
Reading B
Reading C
Reading D
Reading E
Reading F
Reading G
Reading H
Reading I
Reading J
Reading K
Reading L
Reading M
Reading N
Reading O
Reading P
Reading Q
Reading R
Reading S
Reading T
Reading U
Reading V
Reading W
Reading X
Reading Y
Reading Z
Reading [
Reading \
Reading ]
Reading ^
Reading _
Reading `
Reading a
```

```
Reading b
Reading c
Reading d
Reading e
Reading f
Reading g
Reading h
Reading i
Reading j
Reading k
Reading l
Reading m
Reading n
Reading o
Reading p
Reading q
Reading r
Reading s
Reading t
Reading u
Reading v
Reading w
Reading x
Reading y
Reading z
Reading {
Reading |
Reading }
Reading ~
Reading
Reading ▒
Reading ▒
Reading ▒
Reading ▒
Reading ▒
Reading ▒
Reading ▒
TEST PASSED
190031187$
```

Observation and Analysis:

Here the first test writes a series of X of form XXXXXXXX and the same is read below. Hence a mirrored raid feature. Next, we write all the alphabets and symbols and the mirrored raid feature reads all the written alphabets and symbols in same order. Hence both the above tests prove our Mirrored Raid feature by performing tests.