



Department of Computer Science and Engineering

Course Code: CSE 336

Course Title: Operating System Lab

Submitted by:

Name: Mohammad Faiyaz Khan

Registration Number: 2016331011

Name: S.M. Sadiq-Ur-Rahman Shifath

Registration Number: 2016331001

Name: Tirtha Roy

Registration Number: 2016331075

Submitted to:

Ayesha Tasnim

Assistant Professor, CSE, SUST.

1. The **echo** command is used to display a line of text as output. The 'echo hello world' command displays a line of text as output, which is 'hello world'. If we type the command 'man echo' on linux terminal, the following output is showed.

```
File Edit View Search Terminal Help
ECHO(1)                                User Commands
    ECHO(1)

NAME
    echo - display a line of text

SYNOPSIS
    echo [SHORT-OPTION]... [STRING]...
    echo LONG-OPTION

DESCRIPTION
    Echo the STRING(s) to standard output.

    -n      do not output the trailing newline

    -e      enable interpretation of backslash escapes
Manual page echo(1) line 1/71 22% (press h for help or q to quit)
```

2. The **man man** command was typed and the operation of the **man** command was shown as output. The command **echo \$SHELL** shows the name of the login shell that is currently in use. If we type the command on linux terminal, the following output is shown.

```
File Edit View Search Terminal Help
tirtha@tirtha-X556UQK:~/OS$ echo $SHELL
/bin/bash
tirtha@tirtha-X556UQK:~/OS$
```

3.

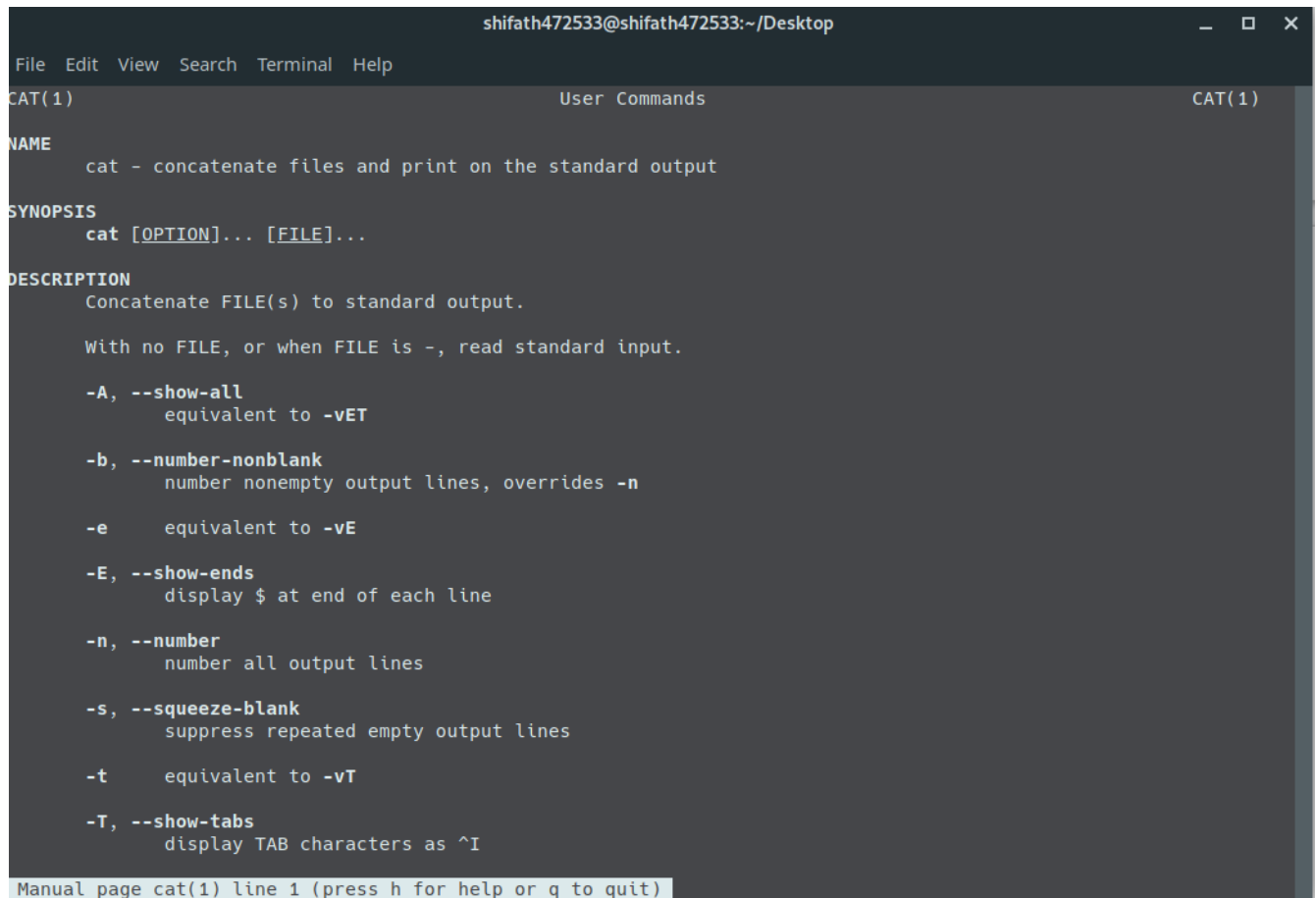
man: man command is used to display the manual of a command or function that we can run on the terminal of linux. It provides a detailed view of the command which includes Name, Synopsis, Description, options, exit, status, return values, errors, files, versions, examples, authors and see also.

Example:

If the following command is typed in linux command shell

```
$ man cat
```

Then the output is



```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
CAT(1) User Commands CAT(1)
NAME
  cat - concatenate files and print on the standard output
SYNOPSIS
  cat [OPTION]... [FILE]...
DESCRIPTION
  Concatenate FILE(s) to standard output.
  With no FILE, or when FILE is -, read standard input.
  -A, --show-all          equivalent to -vET
  -b, --number-nonblank    number nonempty output lines, overrides -n
  -e                      equivalent to -vE
  -E, --show-ends          display $ at end of each line
  -n, --number             number all output lines
  -s, --squeeze-blank      suppress repeated empty output lines
  -t                      equivalent to -vT
  -T, --show-tabs          display TAB characters as ^I
Manual page cat(1) line 1 (press h for help or q to quit)
```

who: This command is used to show the following informations.

1. Login name of the users .
2. Terminal line numbers
3. Login time of the users in to system
4. Remote host name of the user

If **who** is run in the shell in this way

```
$who
```

then output will be

```
shifath472533 :1      2019-03-30 11:12 (:1)
```

If the who command with other attribute is typed in the linux command shell then the following output is displayed.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ man cat
[shifath472533@shifath472533 Desktop]$ man who
[shifath472533@shifath472533 Desktop]$ who
shifath472533 :1          2019-03-30 11:12 (:1)
[shifath472533@shifath472533 Desktop]$ who -m -H
NAME      LINE      TIME      COMMENT
[shifath472533@shifath472533 Desktop]$ who -p -H
NAME      LINE      TIME      PID COMMENT
[shifath472533@shifath472533 Desktop]$ who -T -H
NAME      LINE      TIME      COMMENT
shifath472533 ? :1          2019-03-30 11:12 (:1)
[shifath472533@shifath472533 Desktop]$ who -u
shifath472533 :1          2019-03-30 11:12 ?          17339 (:1)
[shifath472533@shifath472533 Desktop]$ who -b -H
NAME      LINE      TIME      PID COMMENT
system boot 2019-03-30 17:11
[shifath472533@shifath472533 Desktop]$ who -l -H
NAME      LINE      TIME      IDLE      PID COMMENT
[shifath472533@shifath472533 Desktop]$
```

In the command shell different formats of who command is displayed.

cat:

This command reads data from the file and shows their contents as their output. It is used to create,view,concatenate files etc.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ cat file01
It's for LINUX learning purpose . I this is gonna be interesting.
[shifath472533@shifath472533 Desktop]$ cat file01 file02
It's for LINUX learning purpose . I this is gonna be interesting.
This another one.

[shifath472533@shifath472533 Desktop]$ cat -n file01
 1 It's for LINUX learning purpose . I this is gonna be interesting.
[shifath472533@shifath472533 Desktop]$ cat -n file02
 1 This another one.
 2 This is line 2.
 3
 4
[shifath472533@shifath472533 Desktop]$ cat file01 > file02
[shifath472533@shifath472533 Desktop]$ cat file02
It's for LINUX learning purpose . I this is gonna be interesting.
[shifath472533@shifath472533 Desktop]$ cat file01 >> file02
[shifath472533@shifath472533 Desktop]$ cat file02
It's for LINUX learning purpose . I this is gonna be interesting.
It's for LINUX learning purpose . I this is gonna be interesting.
[shifath472533@shifath472533 Desktop]$
```

The output for cat command is displayed below -

cd:

This command is used to change the current working directory.

The output of this command is displayed below.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Pictures]$ cd /
[shifath472533@shifath472533 /]$ cd home
[shifath472533@shifath472533 home]$ ls
lost+found shifath shifath472533
[shifath472533@shifath472533 home]$ cd shifath472533
[shifath472533@shifath472533 ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[shifath472533@shifath472533 ~]$ cd Desktop
[shifath472533@shifath472533 Desktop]$
```

cp:

cp is used to copy files or a group of files or directory.

The output of the cp command is given below

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ cp file01 file02
cp: overwrite 'file02'? Y
[shifath472533@shifath472533 Desktop]$ cat file01
It's for LINUX learning purpose . I this is gonna be interesting.
[shifath472533@shifath472533 Desktop]$ cat file02
It's for LINUX learning purpose . I this is gonna be interesting.
[shifath472533@shifath472533 Desktop]$ cp file01 file02 file03 des
[shifath472533@shifath472533 Desktop]$ ls des
file01 file02 file03
[shifath472533@shifath472533 Desktop]$
```

ps:

This command is used to show the currently running processes with their PIDs and some other details like TTY,TIME,CMD etc.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ ps
  PID TTY          TIME CMD
   661 pts/0        00:00:00 bash
 11089 pts/0        00:00:00 ps
[shifath472533@shifath472533 Desktop]$ ps -A
  PID TTY          TIME CMD
    1 ?           00:00:07 systemd
    2 ?           00:00:00 kthreadd
    3 ?           00:00:00 rcu_gp
    4 ?           00:00:00 rcu_par_gp
    6 ?           00:00:00 kworker/0:0H
    8 ?           00:00:00 mm_percpu_wq
    9 ?           00:00:00 ksoftirqd/0
   10 ?           00:00:01 rcu_preempt
```

Here , ps -a command shows all the running processes.

ls:

This command displays the directory contents of files and directories.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ ls des
file01 file02 file03
[shifath472533@shifath472533 Desktop]$ ls
des file01 file02 file03
[shifath472533@shifath472533 Desktop]$
```

By using 'ls des' command we can see all the files in des directory.

mv:

This command renames a file or folder or moves group of files to different directories.

Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ ls
des file01 file02 file03
[shifath472533@shifath472533 Desktop]$ mv file01 a
[shifath472533@shifath472533 Desktop]$ ls
a des file02 file03
[shifath472533@shifath472533 Desktop]$
```

Here 'file01' is renamed to 'a'.

rm:

This command is used to remove references to objects from the file system.

Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ ls
a des file02 file03
[shifath472533@shifath472533 Desktop]$ rm a
[shifath472533@shifath472533 Desktop]$ ls
des file02 file03
[shifath472533@shifath472533 Desktop]$
```

Here file 'a' is removed.

mkdir:

This command is used to create new directories.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ ls
des file02 file03
[shifath472533@shifath472533 Desktop]$ mkdir newdir
[shifath472533@shifath472533 Desktop]$ ls
des file02 file03 newdir
[shifath472533@shifath472533 Desktop]$
```

Here new directory 'newdir' has been created.

rmdir:

This command is used to remove empty directory. If there is another directory inside the directory then it can't be removed by this command.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ ls
des file02 file03 newdir
[shifath472533@shifath472533 Desktop]$ rmdir newdir
[shifath472533@shifath472533 Desktop]$ ls
des file02 file03
[shifath472533@shifath472533 Desktop]$
```

Here the empty directory 'newdir' has been removed by this command.

echo:

This command displays a line of text or string .

Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ echo operating system
operating system
[shifath472533@shifath472533 Desktop]$ echo -e "operating \bsystem"
operatingsystem
[shifath472533@shifath472533 Desktop]$
```

Here a string “operating system” has been displayed by this command.

More:

This command is used to view large text files to display one screen at a time. It is also used sometimes with some other command after a pipe.

After writing the command “more -d cow” the output is like below. Here ‘cow’ is the text file.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
Cow Essay (400 words)

Cow is a very useful pet animal. It is a successful domestic animal kept by people at home for many purposes. It is a four footed female animal having a large body, two horns, two eyes, two ears, one nose, one mouth, one head, a big back and stomach. She eats large amount of food in one time. She gives us milk to make us healthy and strong. It keeps us away from the diseases and infections by increasing our immunity power. She is a sacred animal and worshipped in India like a Goddess. She has been given a status of Mother in the Hindu society and called as “Gau Mata”.

It is a very famous milk giving animal useful for many purposes. In Hindu religion, it is considered as Gau Dan is the biggest Dan in the world. Cow is a sacred animal to Hindus. Cow gives us lots of benefit all through her life and even after her death. She gives us milk, calf (either female cow or male cow ox), co-dung, gau-mutra while living and lots of leather and strong bones after death. So, we can say that her whole body is useful to us. We can get lots of products from the milk given by her like ghee, cream, butter, curd, dahi, whey, condensed milk, variety of sweets, etc. Her co-dung and urine is highly useful to the farmers for making natural fertilizer for plants, trees, crops, etc.

She eats green grasses, foods, grains, hay and other eatable things. She uses her one pair of strong and tight horns to attack on the people as a defence organ
:
```

less:

This command is used to read contents of a text files one page per time.

If the “dmesg | less” command is typed then the output is as follows:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[ 0.000000] microcode: microcode updated early to revision 0x8e, date = 2018-03-24
[ 0.000000] Linux version 4.19.30-1-MANJARO (builduser@development) (gcc version 8.2.1 20181127 (GCC)) #1 SMP PREEMPT Tue Mar 19 17:49:12 UTC 2019
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.19-x86_64 root=UUID=d066eeba-4bb3-4715-8bde-8d1c4160c549 rw quiet resume=UUID=a00b8e39-a8ba-4587-aac5-787f3a1677dd
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
[ 0.000000] AMD AuthenticAMD
[ 0.000000] Centaur CentaurHauls
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x008: 'MPX bounds registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x010: 'MPX CSR'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] x86/fpu: xstate_offset[3]: 832, xstate_sizes[3]: 64
[ 0.000000] x86/fpu: xstate_offset[4]: 896, xstate_sizes[4]: 64
[ 0.000000] x86/fpu: Enabled xstate features 0x1f, context size is 960 bytes, using 'compacted' format.
[ 0.000000] BIOS-provided physical RAM map:
:
```

date:

This command is used to display system date and time.

Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ date
Sat Mar 30 13:57:19 +06 2019
[shifath472533@shifath472533 Desktop]$
```

time:

This command is used to determine how long a given command takes to run.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ time date
Sat Mar 30 14:03:12 +06 2019

real    0m0.002s
user    0m0.002s
sys     0m0.000s
[shifath472533@shifath472533 Desktop]$
```

Here 'real' shows the total time to execute the call. And 'user' shows time in user mode and 'sys' shows the time in kernel mode.

kill:

This command is used to terminate process manually.

Output:

```
shifath472533@shifath472533:~  
File Edit View Search Terminal Help  
[shifath472533@shifath472533 ~]$ ps  
  PID TTY          TIME CMD  
14107 pts/0    00:00:00 bash  
14370 pts/0    00:00:00 ps  
[shifath472533@shifath472533 ~]$ kill 14370  
bash: kill: (14370) - No such process  
[shifath472533@shifath472533 ~]$
```

By this command process ps with PID - '14370' has been terminated manually.

history:

This command is used to view the previously executed commands.

```
shifath472533@shifath472533:~  
File Edit View Search Terminal Help  
[shifath472533@shifath472533 ~]$ history 10  
148  ps  
149  kill -L  
150  clear  
151  ps  
152  kill 14370  
153  clear  
154  history  
155  history 10  
156  clear  
157  history 10  
[shifath472533@shifath472533 ~]$
```

Here the history of last 10 processes executed has been shown.

chmod:

This command is used to change the permissions of a file or directory.

Example output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ ls
cow des file02 file03
[shifath472533@shifath472533 Desktop]$ chmod u=r file02
[shifath472533@shifath472533 Desktop]$ cat file02
It's for LINUX learning purpose . I this is gonna be interesting.
[shifath472533@shifath472533 Desktop]$ cat file03
This is very much interesting to know about different commnds like cat ,cp etc.
[shifath472533@shifath472533 Desktop]$ cp file03 file02
cp: unwritable 'file02' (mode 0475, r--rwxr-x); try anyway? N
[shifath472533@shifath472533 Desktop]$ chmod u=wrx file02
[shifath472533@shifath472533 Desktop]$ cat file02
It's for LINUX learning purpose . I this is gonna be interesting.
[shifath472533@shifath472533 Desktop]$ cp file03 file02
cp: overwrite 'file02'? Y
[shifath472533@shifath472533 Desktop]$ cat file02
This is very much interesting to know about different commnds like cat ,cp etc.
[shifath472533@shifath472533 Desktop]$
```

Here the user permission has been changed.

chown:

This command changes the owner of a file or dieectory. It is faster and easier than changing permission.

Example Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ ls -l
total 20
-rw-r--r-- 1 shifath472533 shifath472533 6532 Mar 30 13:42 cow
drwxr-xr-x 2 shifath472533 shifath472533 4096 Mar 30 12:44 des
-rwxrwxr-x 1 root shifath472533 80 Mar 30 16:57 file02
-rw-r--r-- 1 shifath472533 shifath472533 80 Mar 30 12:38 file03
[shifath472533@shifath472533 Desktop]$ sudo chown shifath472533 file02
[shifath472533@shifath472533 Desktop]$ ls -l
total 20
-rw-r--r-- 1 shifath472533 shifath472533 6532 Mar 30 13:42 cow
drwxr-xr-x 2 shifath472533 shifath472533 4096 Mar 30 12:44 des
-rwxrwxr-x 1 shifath472533 shifath472533 80 Mar 30 16:57 file02
-rw-r--r-- 1 shifath472533 shifath472533 80 Mar 30 12:38 file03
[shifath472533@shifath472533 Desktop]$
```

Here in the beginning, we can see by “ls -l” command that the owner of the file “file02” is “root”. By |”chown” command we have changed the owner and set to “shifath472533”. Again by “ls -l” command we can see that the owner of “file02” has been changed.

finger :

This command is used to display informations about users.

pwd:

This command print the name of the current working directory.

Example output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ pwd
/home/shifath472533/Desktop
[shifath472533@shifath472533 Desktop]$ type -a pwd
pwd is a shell builtin
pwd is /usr/bin/pwd
[shifath472533@shifath472533 Desktop]$
```

Here, the by this command the current directory is displayed.

cal:

This command is used to display the calendar of a specific month or a whole year.

For only “cal” command the calendar of current month is displayed.

Example output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ cal
      March 2019
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
[shifath472533@shifath472533 Desktop]$
```

Here , by **cal** command the calendar of the current month has been displayed and in 2nd case calendar by **cal 2019** command the calendar of whole year is displayed.

Logout:

This command allows to logout from a session.

shutdown:

This command shuts down the machine. It can be controlled by time in **HH:MM** format or “+m” (here, m is no. of minutes).

Example output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ shutdown
Shutdown scheduled for Sat 2019-03-30 18:56:24 +06, use 'shutdown -c' to cancel.
[shifath472533@shifath472533 Desktop]$
```

Here, after entering shutdown command the pc shuts down after 1 minute.

4. **sed** command is used to delete the first and last character in each line of a file.

Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ cat FILE01
It's for LINUX learning purpose .
This is gonna be interesting.
I love operating system.
It's amazing.
Oh it's awesome.
I'm loving this.
I'm excited about all these things.
[shifath472533@shifath472533 Desktop]$ sed -e "s/\([^ ]*\) *\([^ ]*\)/\2 \1 /g" file01
[shifath472533@shifath472533 Desktop]$ sed 's/.$//; s/^.///' FILE01
t's for LINUX learning purpose .
his is gonna be interesting
love operating system
t's amazing
h it's awesome
'm loving this
'm excited about all these things
[shifath472533@shifath472533 Desktop]$ sed -e "s/\([^ ]*\) *\([^ ]*\)/\2 \1 /g" file0
[shifath472533@shifath472533 Desktop]$
```

grep command is used to find how many lines of a file contain a given word. Here the file name and the word are provided as inputs.

Output:


```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ cat file02
This another one.
This is line 2.
This is just awesome.
This is beautiful.
This is actually not true.
This is actuallyt not this.
But I'm happy.
[shifath472533@shifath472533 Desktop]$ grep -c 'this' file02
1
[shifath472533@shifath472533 Desktop]$ grep -c 'This' file02
6
[shifath472533@shifath472533 Desktop]$ grep -c 'true' file02
1
[shifath472533@shifath472533 Desktop]$ █
```

awk is a scripting language used for manipulating data and generating reports. The **awk** command programming language requires no compiling, and allows the user to use variables, numeric functions, string functions, and logical operators.

By default **awk** prints every line of data from the specified file.

\$ awk '{print}' os.txt

Here, it's printing the value of os.txt.

5.who command was used and redirected to a file myfile1 and more command was used to see the contents of myfile1. Output is given below.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ who
shifath472533 :1          2019-04-03 17:28 (:1)
[shifath472533@shifath472533 Desktop]$ who > myfile1 | more myfile1
[shifath472533@shifath472533 Desktop]$ █
```

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help

shifath472533 :1          2019-04-03 17:28 (:1)
myfile1 (END)
```

6. The **date** and **who** command was used in sequence such that the output of date is displayed and the output of who command is redirected to a file myfile2 and more command was used to display myfile2. Output is given below:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ who|date
Sat Mar 30 20:05:57 +06 2019
[shifath472533@shifath472533 Desktop]$ who > myfile2|date
Sat Mar 30 20:07:00 +06 2019
[shifath472533@shifath472533 Desktop]$ more myfile2
[shifath472533@shifath472533 Desktop]$
```

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help

shifath472533 :1          2019-03-30 11:12 (:1)
myfile2 (END)
```


7. **sed** command is used to swap the first and second words in each line in a file. Output is given below.

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ cat file01
It's for LINUX learning purpose .
This is gonna be interesting.
I love operating system.
It's amazing.
[shifath472533@shifath472533 Desktop]$ sed -e "s/\([^ ]*\) *\([^ ]*\)/\2 \1 /g" file01
for It's LINUX learning purpose .
is This gonna be interesting.
love I operating system.
amazing. It's
[shifath472533@shifath472533 Desktop]$
```

8. A shell script program and a c program has been written to display hello world and their time is compared.

```
Open  +  hell...  Save  ~/Des...
#include<stdio.h>

int main()
{
    printf("HELLO WORLD\n");
    return 0;
}
```

Output is given below

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ gcc hello.c
[shifath472533@shifath472533 Desktop]$ gcc -o hello hello.c
[shifath472533@shifath472533 Desktop]$ ./hello
HELLO WORLD
[shifath472533@shifath472533 Desktop]$ time ./hello
HELLO WORLD

real    0m0.004s
user    0m0.000s
sys     0m0.004s
[shifath472533@shifath472533 Desktop]$ time sh test.sh
HELLO WORLD

real    0m0.020s
user    0m0.011s
sys     0m0.009s
[shifath472533@shifath472533 Desktop]$
```

9. A shell script is written which takes a command line as argument and reports whether it is directory, a file or something else. Output is given below

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ sh task09.sh
enter file
file01
file exists n it is an ordinary file
[shifath472533@shifath472533 Desktop]$ sh task09.sh
enter file
des
directory file
[shifath472533@shifath472533 Desktop]$ ls
a.out 'Empty File' file03 hello_world newfile
cow file01 hello myfile1 task09.sh
des file02 hello.c myfile2 test.sh
[shifath472533@shifath472533 Desktop]$
```

10. A shell script is written that determines the period for which a specified user is working on the system is given below

```
Task10.sh
~/Desktop
task10.sh x
#!/bin/bash
if [ $# -eq 0 ]
then
    echo "Enter File Name : "
else
    for i in $*
    do
        if [ -f $i ]
        then
            a=`echo $i | tr '[a-z]' '[A-Z]'`
            mv $i $a
            echo "new file name: $a"
        else
            echo "FileName $i does not exist"
        fi
    done
fi
```

output :

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ ls
a.out      file01  hello.c    myfile3    task10.sh  task13.sh
cow        file02  hello_world newfile     Task10.sh  task14.sh
des        file03  myfile1    newline    task11.sh  test.sh
'Empty File' hello  myfile2    task09.sh  task12.sh  upper.sh
[shifath472533@shifath472533 Desktop]$ sh task10.sh file01
new file name: FILE01
[shifath472533@shifath472533 Desktop]$
```

11. A shell script is written that determines the period for which a specified user is working on the system is given below

```
per.sh
~/
echo ENTER USERNAME
read $username
last $username
```

Output:

```
tirtha@tirtha-X556UQK: ~/OS
File Edit View Search Terminal Help
tirtha@tirtha-X556UQK:~/OS$ sh per.sh
ENTER USERNAME
per.sh: 2: read: arg count
tirtha :0 :0 Tue Jul 2 14:40 still logged in
reboot system boot 4.15.0-54-generi Tue Jul 2 14:39 still running
tirtha :0 :0 Tue Jul 2 03:54 - down (02:03)
reboot system boot 4.15.0-54-generi Tue Jul 2 03:51 - 05:57 (02:06)

wtmp begins Tue Jul 2 03:51:27 2019
tirtha@tirtha-X556UQK:~/OS$
```

12. A shell script is written that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers is given below

```
per.sh
~/
FILE=$1
start=$2
end=$3

if[! -f $FILE];then
    echo "$FILE is not a valid file"
    exit 1
fi

str='wc -l $FILE'
array=( $str )
totalLines=${array[0]}

if[[ $end -le $totalLines && $totalLines -gt 0 ]];then
    sed -n "${start},${end}p" $FILE
else
    echo "Range out of bounds"
fi
```

Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ cat FILE01
It's for LINUX learning purpose .
This is gonna be interesting.
I love operating system.
It's amazing.
Oh it's awesome.
I'm loving this.
I'm excited about all these this things.
sed -e "s/\([^ ]*\) *\([^ ]*\)/\2 \1 /g" file01
[shifath472533@shifath472533 Desktop]$ sh task12.sh FILE01 2 5
This is gonna be interesting.
I love operating system.
It's amazing.
Oh it's awesome.
[shifath472533@shifath472533 Desktop]$
```

13. A shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it is given below

```
word=$1
cnt=0

for i in $*
do
    if [ $cnt -eq 0 ]
    then
        cnt=1
        continue
    elif [ -f $i ]
    then
        echo "After deleting lines containing '$word' in '$i' "
        grep -v "$word" $i
    else
        echo "FileName $i does not exist"
        continue
    fi
done
```

Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ cat FILE02
This another one.
This is line 2.
This is just awesome.
This is beautiful.
This is actually not true.
This is actuallyt not this.
But I'm happy.

[shifath472533@shifath472533 Desktop]$ sh task13.sh This FILE02
After deleting lines containing 'This' in 'FILE02'
But I'm happy.

[shifath472533@shifath472533 Desktop]$ █
```

14. A shell script is written that extract a sub-string from a given string is given below

```
task14.1.sh
:
echo "Enter string :"
read str
len=$(echo ${#str})
echo "Length of the given string is " $len
```

Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ sh task14.1.sh
Enter string :
Operating system is very much interesting.
Length of the given string is 42
[shifath472533@shifath472533 Desktop]$
```

A shell script that find the length of a given string is given below

```
task14.2.sh
~/Desktop
Open ▾
echo "Enter string : "
read str
echo "Enter the starting position of the substring:"
read pos
echo "Enter the length of the substring :|"
read len
st=$(echo ${str:pos:len})
echo "The required substring is : '$st'"
```

Output:

```
shifath472533@shifath472533:~/Desktop
File Edit View Search Terminal Help
[shifath472533@shifath472533 Desktop]$ sh task14.2.sh
Enter string :
Operating system is very much interesting.
Enter the starting position of the substring:
3
Enter the length of the substring :
15
The required substring is : 'rating system i'
[shifath472533@shifath472533 Desktop]$
```