

Operating Systems

Lab File

Name – Tushya Gupta
Roll number – UE233106
Group – 6

Index

S. no.	Name	Page
1	Basic Commands	3-10
2	Shell Scripting	11-14
2.1	Add 2 numbers	11
2.2	Calculator	11-12
2.3	Swapping 2 numbers	12-13
2.4	Factorial	13
2.5	Fibonacci	13-14
3	Grep command	15-16
4	Sed Command	17-18

Basic Commands

- Man – Give information of a command and what various flags do

```
MAN(1)                                Manual pager utils                                MAN(1)
NAME
  man - an interface to the system reference manuals
SYNOPSIS
  man [man options] [[section] page ...] ...
  man -k [apropos options] regexp ...
  man -K [man options] [section] term ...
  man -f [whatis options] page ...
  man -l [man options] file ...
  man -w|-W [man options] page ...
```

- Date - shows system time and date

```
> date
Wed Apr 23 10:29:41 PM IST 2025
```

- Cal/ncal – show the calendar for a specified month and year

```
> cal
      April 2025
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
```

- Bc – turns on calculator mode

```
> bc
bc 1.08.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2018, 2024
Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
5+12
17
2*50
100
```

- Cd – change directory

```
.../college/os/shell_scripting on 🐙 main +/- ! 18s  
> cd ..  
  
.../programs/college/os on 🐙 main +/- !  
> 
```

- Ls – list contents of directory

```
.../programs/college/os on 🐙 main +/- !  
> ls  
📁 shell_scripting
```

- Who/whoami – show current logged in user

```
> whoami  
tushya
```

- Clear – clear the terminal


```
.../programs/college/os on 🐙 main +/- ! [0/0]  
> 
```





- Mkdir – create a new directory


```
.../programs/college/os on 🐙 mai  
> mkdir newdir  
  
.../programs/college/os on 🐙 mai  
> ls  
📁 newdir 📁 shell_scripting
```




- Rm/rmdir – remove a file or a directory

```
.../programs/college/os on  main +/
> rmdir newdir/


.../programs/college/os on  main +/
> ls
 shell_scripting
```










- Mv – move a file or directory

```
.../programs/college/os on  main +/
> mv shell_scripting/fib.sh .

.../programs/college/os on  main +/
> ls
 fib.sh  shell_scripting
```

- Cp – copy a file or directory

```
.../programs/college/os on  main +/
> cp fib.sh shell_scripting/fib.sh

.../programs/college/os on  main +/
> lt
 .
├──  fib.sh
└──  shell_scripting
    ├──  add.sh
    ├──  calculator.sh
    ├──  factorial.sh
    ├──  fib.sh
    └──  swapping.sh
```

- Echo – output text or variables onto the terminal

```
> echo "hello world"
hello world
```

- Printf – like echo but provides more output format control using C like format specifiers

```
> printf "%s %s!" "Hello" "World"
Hello World!↵
```

- Ifconfig – used to configure and display network interface parameters

```
> ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
inet 127.0.0.1 netmask 0xff000000
inet6 ::1 prefixlen 128
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
nd6 options=201<PERFORMNUD,DAD>
```

- Vi/nano – terminal based text editors

```
GNU nano 8.4
#!/usr/bin/env bash
echo -n "Enter ending number: "
read a

b=0
c=1

while [ $b -le $a ]; do
    echo $b
    temp=$b
    b=$c
    c=$((b+$temp))
done

^G Help      ^O Write Out  ^F Where Is   ^_
^X Exit      ^R Read File  ^\ Replace
```

```
#!/usr/bin/env bash
echo -n "Enter ending number: "
read a

b=0
c=1

while [ $b -le $a ]; do
    echo $b
    temp=$b
    b=$c
    c=$((b+$temp))
done
```

- Pwd – print the absolute path of current working directory

```
../programs/college/os on 🐚 main +/- !?
> pwd
/home/tushya/programs/college/os
```


- Su – used to switch user

```
> su newuser
su: user newuser does not exist or the user entry does not contain all the required fields
```

- Kill – used to kill a process by giving its pid

```
> kill 12432
kill: sending signal to 12432 failed: No such process
```

- Ping – send packets to a url or ip address

```
> ping google.com
PING google.com (142.250.206.110): 56 data bytes
64 bytes from 142.250.206.110: icmp_seq=0 ttl=118 time=50.251 ms
64 bytes from 142.250.206.110: icmp_seq=1 ttl=118 time=47.820 ms
64 bytes from 142.250.206.110: icmp_seq=2 ttl=118 time=42.818 ms
64 bytes from 142.250.206.110: icmp_seq=3 ttl=118 time=53.382 ms
64 bytes from 142.250.206.110: icmp_seq=4 ttl=118 time=43.454 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 42.818/47.545/53.382/4.014 ms
```

- Traceroute – trace the path to a url or ip address

```
> traceroute google.com
traceroute to google.com (142.250.206.110), 64 hops max, 40 byte packets
 1  unit (192.168.1.1)  9.220 ms  3.077 ms  2.946 ms
 2  10.240.15.125 (10.240.15.125)  28.542 ms  28.014 ms  25.245 ms
 3  172.30.1.138 (172.30.1.138)  37.993 ms  26.128 ms
   172.30.1.154 (172.30.1.154)  29.941 ms
 4  125.16.191.65 (125.16.191.65)  37.042 ms
   125.16.191.81 (125.16.191.81)  33.994 ms
   125.16.191.65 (125.16.191.65)  31.434 ms
 5  116.119.52.34 (116.119.52.34)  41.143 ms
   182.79.208.12 (182.79.208.12)  28.908 ms
   116.119.52.34 (116.119.52.34)  47.718 ms
 6  142.250.161.56 (142.250.161.56)  33.223 ms  56.612 ms  36.917 ms
 7  * * *
 8  142.251.52.214 (142.251.52.214)  59.018 ms
   142.251.52.228 (142.251.52.228)  36.584 ms
   142.251.76.168 (142.251.76.168)  44.772 ms
 9  142.251.76.193 (142.251.76.193)  62.100 ms  51.188 ms
   142.251.76.195 (142.251.76.195)  46.289 ms
10  192.178.82.235 (192.178.82.235)  44.452 ms
   142.250.63.53 (142.250.63.53)  59.764 ms  146.436 ms
11  de111s20-in-f14.1e100.net (142.250.206.110)  49.023 ms  49.337 ms  38.160 ms
```

- chmod – Change the permissions on a file

```
.../programs/college/os on ? main +/- !?
> chmod +x fib.sh

.../programs/college/os on ? main +/- !?
> l
.rwxr-xr-x 144 tushya 23 Apr 21:45 -N fib.sh
```

- chown – Change the owner of a file or directory

```
.../programs/college/os on ? main +/- !?  
> sudo chown root fib.sh  
[sudo] password for tushya:  
  
.../programs/college/os on ? main +/- !?  
> ls  
-> fib.sh  shell_scripting  
  
.../programs/college/os on ? main +/- !?  
> l  
.rwxr-xr-x 144 root  23 Apr 21:45 -N -> fib.sh
```

- tar – To compress and decompress files into .tar

```
.../programs/college/os on ? main +/- !  
> tar -cvf shell_scripting.tar shell_scripting/  
shell_scripting/  
shell_scripting/calculator.sh  
shell_scripting/swapping.sh  
shell_scripting/fib.sh  
shell_scripting/add.sh  
shell_scripting/factorial.sh  
  
.../programs/college/os on ? main +/- !?  
> ls  
- shell_scripting  - shell_scripting.tar
```

- shutdown – Shutdown computer
- cut – Get specific selections from a file
- free – Shows the memory usage

```
.../programs/college/os on ? main +/- !?  
> free
```

	total	used	free	shared	buff/cache	av
available						
Mem:	16316668	4126120	10359592	155080	2285448	1
2190548						
Swap:	4194300	0	4194300			

- tr – translate characters from input to another characters

```
> echo "hello 123" | tr 'a-z' 'A-Z'  
HELLO 123
```


- comm – segregates common lines between files

```
.../programs/college/os on ? main +/- !?  
> comm shell_scripting/fib.sh shell_scripting/factorial.sh  
#!/usr/bin/env bash  
  
    echo -n "Enter a number: "  
echo -n "Enter ending number: "  
    read a  
comm: file 1 is not in sorted order  
comm: file 2 is not in sorted order  
  
b=0  
    b=1  
c=1  
  
    for i in $(seq 1 $a); do  
        b=$((b*$i))  
    done  
  
    echo $b  
while [ $b -le $a ]; do  
    echo $b  
    temp=$b  
    b=$c  
    c=$((b+$temp))  
done
```

- type – Shows how a command is interpreted by the shell

```
> type git  
git is /usr/bin/git
```

- find – Find files and directories in a hierarchy

```
> find **/fib.sh  
shell_scripting/fib.sh
```

- cat – display the contents of a file

```
> cat shell_scripting/fib.sh
#!/usr/bin/env bash

echo -n "Enter ending number: "
read a

b=0
c=1

while [ $b -le $a ]; do
    echo $b
    temp=$b
    b=$c
    c=$((b+temp))
done
```

- touch – Update timestamp of a file

```
~/programs/college/os on 🐚 main +/- !?
> touch newfile

~/programs/college/os on 🐚 main +/- !?
> ls
new  newfile  shell_scripting  shell_scripting.tar
```

Shell Scripting

Q1 - Add 2 numbers

Code –

```
#!/usr/bin/env bash
```

```
echo -n "Give first number: "
```

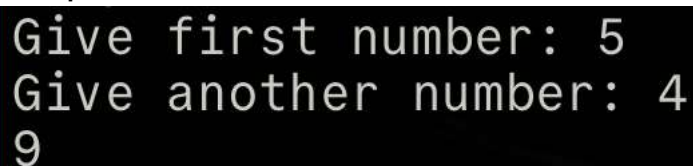
```
read a
```

```
echo -n "Give another number: "
```

```
read b
```

```
echo $((($a+$b));
```

Output –



```
Give first number: 5
Give another number: 4
9
```

Q2 – Calculator

Code –

```
#!/usr/bin/env bash
```

```
echo -n "Give first number: "
```

```
read a
```

```
echo -n "Give operator: "
```

```
read op
```

```
echo -n "Give another number: "
```

```
read b
```

```
case $op in
```

```
"+" )
```

```
    echo $((($a+$b));
```

```
;;

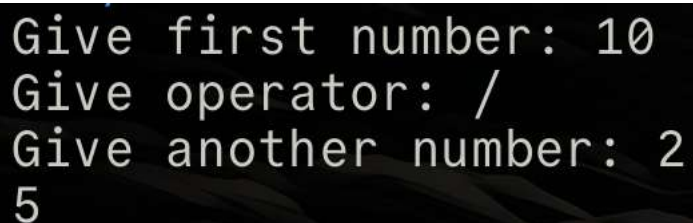
")
echo $((($a-$b));
;;

"*)
echo $((($a*$b));
;;

"/")
echo $((($a/$b));
;;
```

Esac

Output –



```
Give first number: 10
Give operator: /
Give another number: 2
5
```

Q3 – Swapping

Code –

```
#!/usr/bin/env bash
```

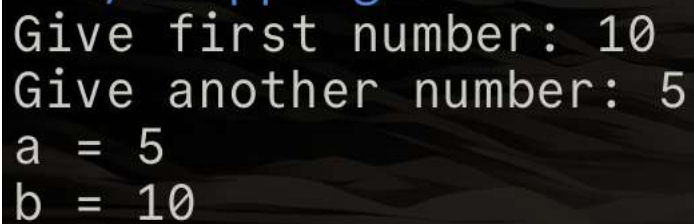
```
echo -n "Give first number: "
read a
```

```
echo -n "Give another number: "
read b
```

```
temp=$a
a=$b;
b=$temp;
```

```
echo "a = $a";  
echo "b = $b";
```

Output –



```
Give first number: 10  
Give another number: 5  
a = 5  
b = 10
```

Q4 – Factorial

Code –

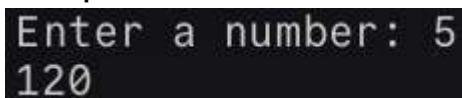
```
#!/usr/bin/env bash
```

```
echo -n "Enter a number: "  
read a
```

```
b=1  
for i in $(seq 1 $a); do  
    b=$((b*$i))  
done
```

```
echo $b
```

Output –



```
Enter a number: 5  
120
```

Q5 – Fibonacci series

Code –

```
#!/usr/bin/env bash
```

```
echo -n "Enter ending number: "
```



```
read a
```

```
b=0
```

```
c=1
```

```
while [ $b -le $a ]; do
```

```
    echo $b
```

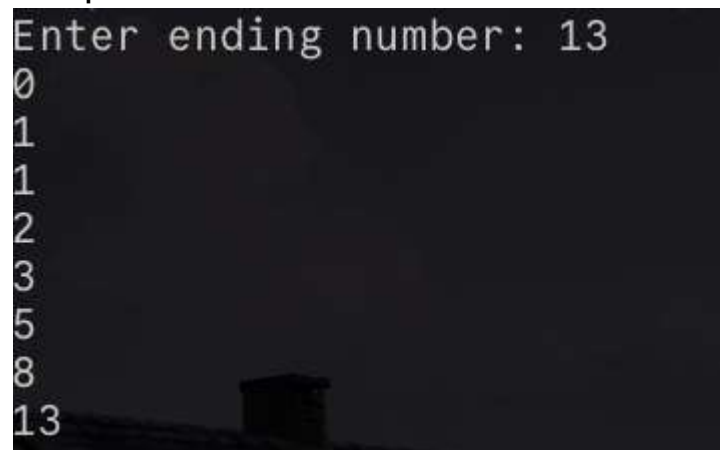
```
    temp=$b
```

```
    b=$c
```

```
    c=$((b+$temp))
```

```
done
```

Output –



```
Enter ending number: 13
0
1
1
2
3
5
8
13
```

Grep Command

The **grep** command in Linux is a powerful text-search utility. It scans files or input for lines that match a pattern — typically using regular expressions — and prints those lines.

```
> grep read calculator.sh
read a
read op
read b
```

Flag	Description
-i	Case-insensitive search
-v	Invert match — show lines not matching the pattern
-r or -R	Recursively search directories
-l	List only filenames with matches
-L	List files without matches
-n	Show line numbers of matching lines
-c	Count matching lines (per file)
-o	Show only the matching part of the line
-w	Match whole words only (e.g., match "cat" but not "catalog")
-x	Match whole lines
-A N	Show N lines after each match
-B N	Show N lines before each match
-C N	Show N lines before & after each match (context)
-e	Use multiple search patterns
-f file	Read patterns from a file

```
> grep -A 2 read calculator.sh  
read a
```

```
echo -n "Give operator: "  
read op
```

```
echo -n "Give another number: "  
read b
```

```
case $op in
```

```
> grep -l read *  
add.sh  
calculator.sh  
factorial.sh  
fib.sh  
swapping.sh
```

```
> grep -L for *  
add.sh  
calculator.sh  
fib.sh  
swapping.sh
```

```
> grep -w number: *  
add.sh:echo -n "Give first number: "  
add.sh:echo -n "Give another number: "  
calculator.sh:echo -n "Give first number: "  
calculator.sh:echo -n "Give another number: "  
factorial.sh:echo -n "Enter a number: "  
fib.sh:echo -n "Enter ending number: "  
swapping.sh:echo -n "Give first number: "  
swapping.sh:echo -n "Give another number: "
```

Sed Command

sed stands for **Stream Editor** — it's a command-line tool used to **parse and transform text**, especially useful in shell scripting, data cleaning, and batch edits.

It works line-by-line on input (from a file or piped data), applying **editing commands** like substitute, delete, insert, and more.

Flag/Pattern	Description
-n	Suppress automatic printing of pattern space (used with p to print manually)
-e SCRIPT	Add a script to the commands to be executed
-f scriptfile	Read the sed script from a file
-i	Edit files in-place (overwrite the file)
-i.bak	Edit in-place and create a backup with .bak extension
s/pattern/replacement/	Substitution command: replaces pattern with replacement
s/pat/repl/g	Replace all matches on the line, not just the first
s/pat/repl/i	Case-insensitive replacement
s/pat/repl/2	Replace second occurrence only
p	Print the line (often used with -n)
d	Delete the line
q	Quit sed early
a\ text	Append text after the current line
i\ text	Insert text before the current line
c\ text	Change (replace) the entire line with new text

Flag/Pattern	Description
=	Print the current line number
!	Negate a command (e.g., 2!d keeps only line 2)
,	Used for range of lines (e.g., 1,3d deletes lines 1 to 3)

```
> sed 's/read/reed/' fib.sh
#!/usr/bin/env bash

echo -n "Enter ending number: "
reed a

b=0
c=1

while [ $b -le $a ]; do
    echo $b
    temp=$b
    b=$c
    c=$((b+$temp))
done
```

```
> sed '/^$/d' fib.sh
#!/usr/bin/env bash
echo -n "Enter ending number: "
read a
b=0
c=1
while [ $b -le $a ]; do
    echo $b
    temp=$b
    b=$c
    c=$((b+$temp))
done
```

```
> sed -n '/read/p' calculator.sh
read a
read op
read b
```

```
> sed '1,4!d' fib.sh
#!/usr/bin/env bash

echo -n "Enter ending number: "
read a
```

```
> gsed '3i\wow' fib.sh
#!/usr/bin/env bash

wow
echo -n "Enter ending number: "
read a

b=0
c=1

while [ $b -le $a ]; do
    echo $b
    temp=$b
    b=$c
    c=$((b+$temp))
done
```