

INTRODUCTION to LINUX

Linux stands for Lovable Intellect Not Using XP. It was developed by Linus Torvalds and named after him. It is an open-source and community-developed operating system for computers, servers, mainframes, mobile devices and embedded devices. Linux receives requests from system programs and it relays them into computer hardware. Linux is flexible, reliable, secure and it has large community user's support. Linux is compatible with almost every possible file format and can be run on a large no. of devices.

HISTORY

Linux is named after Linus Torvalds. He wrote the first Linux Kernel in 1991.

The resulting Linux kernel has been marked by constant growth throughout its history. Since the initial release of its source code in 1991, it was grown from a small no. of C files under a license prohibiting commercial distribution to the 4.15 version in 2018 with more than 23.3 million lines of source code.

Characteristics of Linux.

- Linux can work on different types of hardware, that's why Linux is 'portable'.
- Linux is open source, so it is free to use and users can also work on enhancing features of the Linux Operating System.
- Linux is secure as it provides password protection and encryption of data.
- Linux is multi-programming as multiple application can be run at same time.

Advantages of Linux.

- Linux is compatible with large number of file format.
- Linux is free of cost so it is easily available for users to download and use.
- Linux is a system that rarely crashes as they are very stable.
- Linux does not collect much user data, so it ensures privacy of users.

Disadvantages of Linux.

- Linux does not provide some hardware drivers which is drawback of linux.
- Some graphic tools are not available for Linux OS.

Teacher's Signature : _____

KERNEL

The Linux Kernel is the main component of a Linux operating system and it is the core interface between a computer's hardware and its processes. It communicates between the 2, managing resources as efficiently as possible.

The kernel is so named because - like a seed inside a hard shell - it exists within the OS and controls all the major functions of the hardware, whether it's a phone, laptop, server, or any other kind of computers.

The Kernel has 4 Jobs :

- (i) Memory Management
- (ii) Process Management
- (iii) Device Drivers
- (iv) System calls, and Security

SHELL

The shell can be defined as a command interpreter within an OS like Linux / GNU or Unix.

It is a program that runs other programs. The shell facilitates every user of the computer as an integer to the Unix. Hence, the user can execute different tools / utilities or commands with a few input data. The shell sends result to the user over the screen when it has completed

Teacher's Signature : _____

running a program which is common output device. That's why it is known as Command Interpreter. The shell is just not a command interpreter also the shell is a programming language with complete construct of a programming language such as function, variable, loops, conditional execution and many others.

VI Editor :

The vi editor is elaborated as Visual Editor. It is available in all Linux distros. It is user-friendly and works same on different distros and platforms. It is very powerful application. An improved version of vi editor is vim. The vi editor has 2 modes -

[i] Command Mode : In this, actions are taken on file. The vi editor starts in this mode. Here, typed words will act as command in vi editor. To pass a command we need to be in command mode.

[ii] Insert Mode : In this, entered text will be inserted into file. The Esc key will take you to command mode from insert mode.

By default, the vi editor starts in command mode. To enter text, you have to be in insert mode, just type 'i' and we will be in insert mode. To exit from insert mode

Teacher's Signature : _____

press Esc key, you will be directed to command mode. If we are not sure which mode you are in, press Esc key twice and you will be in command mode.

VI Editing Commands :

i	-	insert at cursor
a	-	write after cursor
A	-	write at the end of line
Esc	-	terminate insert mode
u	-	undo last change
U	-	undo all changes to entire line
o	-	open a new line
dd	-	delete line
3 dd	-	delete 3 lines
D	-	delete contents of line after the cursor.
C	-	D + insert new text. Press Esc key to end insertion
dw	-	delete word
4 dw	-	delete 4 words
cw	-	change word
x	-	delete character at cursor
r	-	replace character
R	-	override character from cursor onward
s	-	substitute one character under cursor continue to insert.
S	-	substitute entire line and begin to insert at the beginning of line.
s	-	change case of individual character.

Teacher's Signature : _____

Moving within a file :

- k - moves upwards
- j - moves downwards
- h - moves leftwards
- l - moves rightwards

You need to be in command mode to move within a file. The default keys for navigation are mentioned else; you can also use arrows keys on keyword.

Saving & Closing the file :

- shift + z - save the file and quit.
- :w - save the file but keep it open.
- :q - quit without saving.
- :wq - save the file and quit.

J. elyod Sunith

COMMANDS :

1. Cat :

→ to create a file

Syntax - \$ cat > filename

Example - \$ cat > f1

→ to show content of file

Syntax - \$ cat filename

Example - \$ cat f1

→ to copy content of 2 files into another file

Syntax - \$ cat file1 file2 >> newfile_name

Example - \$ cat f1 f2 >> f3

2. ls : show list of directory and files

Syntax - \$ ls

\$ ls -a : show hidden files list

\$ ls -l : show long list

\$ ls -lh : show long list human readable

\$ ls -t : time of file or directory when created

\$ ls -lt : sort the list

\$ ls -i : display all content with inod number

\$ ls -s : shows the size of file

\$ ls -txt : display list of text file

\$ ls -d* : show all the directories

Teacher's Signature : _____

3. **Touch** : to create an empty file

Syntax : \$touch filename

Example : \$ touch a1

→ to change file access and modification time

Syntax : \$touch -a filename

→ to modify time of a file

Syntax : \$touch -m filename

4. **rm** : to delete or remove a file

Syntax : \$ rm filename

Example : \$ rm f2

5. **mv** : to move a file from one directory to another

Syntax : \$ mv filename directoryname

Example : \$ mv f1 d1

→ to rename a file or directory

Syntax : \$ mv oldname newname

Example : \$ mv f1 m1

6. **cp** : to copy a file or directory

Syntax : \$ cp oldname newname

Example : \$ cp f1 f3

→ to copy in another directory

Syntax : \$ cp filename directoryname

7. Sort :

→ to sort a file [by default in ascending order]

Syntax : \$ sort filename

→ to sort a file in descending order

Syntax : \$ sort -r filename

→ to sort file numerically

Syntax : \$ sort -n filename

8. Pipe (|) : used to connect two or more commands

Syntax : \$ command1 | command2

→ to show pagewise details

Syntax : \$ cat f1 | Pg

9. find : to search file or directory

Syntax : \$ find [path] -name [filename]

Example : \$ find . / -name f1

10. locate : to search filename or directory

(works on root user)

Syntax : \$ locate filename

Teacher's Signature : _____

→ to count the no. of files (case-sensitive)

Syntax : \$ locate -c filename

→ to count number of file with specific filename
(case-insensitive)

Syntax : \$ locate -i -c filename

11. mkdir :

→ to make a directory

Syntax : \$ mkdir directory_name

Example : \$ mkdir d1

→ to make multiple directory

Syntax : \$ mkdir directory_name1 / directory_name2 / ...

→ to create a sub-directory

Syntax : \$ mkdir . directory_name

12. cd : used to change current working directory

Syntax : \$ cd directory_name

→ to change current directory to home directory

Syntax : \$ cd ~

Teacher's Signature : _____

→ to change current directory to root directory.
Syntax : \$ cd /

13. rmdir : to delete a directory

Syntax : \$ rmdir directory-name

Example : \$ rmdir d1

14. chown : to change ownership of a file

Syntax : \$ chown filename newusername

Example : \$ chown f1 Sakshi
\$ ls -l

15. ps : display information about currently active process

Syntax : \$ ps

16. kill : to terminate the process

Syntax : \$ kill PID

17. history : show all commands executed on terminal from last session.

Syntax : \$ history

18. `pwd` : show the name of current working directory
Syntax : `$ pwd`

19. `whoami` : display effective user id of root user.
Syntax : `$ whoami`

20. `man` : view the system reference manual
Syntax : `$ man [options]`

→ to display of manual of `ls` command
Syntax : `$ man ls`

21. `tty` : display type of terminal where current command is running
Syntax : `$ tty`

22. `head` : shows the first 10 lines of any file
Syntax : `$ head filename`

→ to show specified number of line
Syntax : `$ head -n filename`

23. `tail` : shows bottom 10 lines of any file
Syntax : `$ tail filename`

Teacher's Signature : _____

→ to show specified number of line
Syntax : \$tail -n filename

24. cal : to display the calendar (of current month with current day highlighted)
Syntax : \$cal

→ to display specific month of a specific year
Syntax : \$cal month year

25. tar -xvf : to create cvf file compressed file to original file.

26. gzip d1.tar : to compress the file.
\$ gzip d1.tar ↴

27. gunzip : to decompress the file
Syntax : \$ gunzip d1.tar.gz ↴

28. chmod : to take permission

1st Method -

\$ chmod 700 fi

2nd Method -

\$ chmod rwx rx -

→ privileges are given as -

- = update
- + add
- minus

\$ chmod o+wx f ↴

\$ chmod g+rwx f ↴

\$ chmod g-rwx f ↴

\$ chmod ug=rwx f ↴

\$ chmod ugo-rwx f ↴

\$ chmod u=r , g=rw , o=rx f ↴

\$ chmod u-wx , g+wx , o=rx f ↴

Teacher's Signature :

Arithmetic

Write a shell script to perform Arithmetic operation.

```
$vi f1.sh
#!/bin/bash
echo "Enter 2 numbers"
read n1 n2
echo sum is $(($n1+$n2))
echo difference is $(($n1-$n2))
echo product is $(($n1*$n2))
echo division is $(($n1/$n2))
echo remainder is $(($n1%$n2))

# chmod +x f1.sh
$ ./f1.sh
```

Output

Enter 2 numbers
500
10
sum is 510
difference is 490
product is 5000
division is 50
remainder is 0

File Management

Open - Read - Write - Append

Open - Read - Write - Append

Write a shell script to show the use of 'expr' keyword

```
#!/bin/bash
read -p "Enter no." n1
read -p "Enter no." n2
echo $(expr $n1 + $n2)
```

Enter no.	100	$54 + 100$	154
Enter no.	(200 - 10) / 2	$(200 - 10) / 2$	90
	300	$10 * 300$	3000

Write a shell script to add two floating point values

```
#!/bin/bash
read -p "Enter no. " n1
read -p "Enter no. " n2
echo $n1+$n2 | bc
```

Output
Enter no. 3.8 Enter no. 4.2 8.0

Output
3.8 + 4.2 = 8.0

Entered number is 50
Number is positive

Write a shell script to check whether entered no. is positive or negative.

```
#!/bin/bash
read -p "Enter no." n
if (( $n >= 0 ))
then
    echo "Number is positive"
else
    echo "Number is negative"
fi
```

Enter no. -50
Number is negative

Write a shell script to check whether entered no. is even or odd.

```
#!/bin/bash
read -p "Enter no." n
if (( $n%2==0 ))
then
echo "Number is Even"
else
echo "Number is Odd"
fi
```

Output
Enter no. 20
Number is Even

Expt. No.

Write a shell script to find greater no. from 3 numbers.

```
#!/bin/bash
read -p "Enter no. " n1
read -p "Enter no. " n2
read -p "Enter no. " n3
if (( $n1 > $n2 )) && (( $n1 > $n3 ))
then
```

```
echo n1 is greater $n1
elif (( $n2 > $n3 ))
then
echo n2 is greater $n2
else
echo n3 is greater $n3
fi
```

```
Enter no. 45
Enter no. 54
Enter no. 20
Output n3 is greater
```

Write a shell script to enter choice between 1-4 and perform Arithmetic Operations.

```
#!/bin/bash
read -p "Enter no." n1
read -p "Enter no." n2
read -p "Enter choice from 1-4 " c
if (( $c == 1 ))
then
echo sum is $(( n1+n2 ))
elif (( $c == 2 ))
then
echo difference is $(( n1-n2 ))
elif (( $c == 3 ))
then
echo product is $(( n1*n2 ))
elif (( $c == 4 ))
then
echo division is $(( n1/n2 ))
else
echo wrong choice
fi
```

```
#!/bin/bash
read -p "Enter no." n1
read -p "Enter no." n2
read -p "Enter choice from 1-4 " c
if (( $c == 1 ))
then
echo sum is $(( n1+n2 ))
elif (( $c == 2 ))
then
echo difference is $(( n1-n2 ))
elif (( $c == 3 ))
then
echo product is $(( n1*n2 ))
elif (( $c == 4 ))
then
echo division is $(( n1/n2 ))
else
echo wrong choice
fi
```

Write a shell script for switch case

```
#!/bin/bash
read -p "Enter choice between 1-7 " a
case $a in
```

```
[1])
```

```
echo Day is Monday
```

```
[2])
```

```
echo Day is Tuesday
```

```
[3])
```

```
echo Day is Wednesday
```

```
[4])
```

```
echo Day is Thursday
```

```
[5])
```

```
echo Day is Friday
```

```
[6])
```

```
echo Day is Saturday
```

```
[7])
```

```
echo Day is Sunday
```

```
[*)
```

```
echo You have entered wrong choice $a
```

```
esac
```

Output

Day is Thursday

Enter choice between 1-7 7

echo Day is Thursday

Write a shell script to make a list using for loop

```
#!/bin/bash
for a in Anushka Sakshi Mahima Shubham
do
echo $a
```

(E)

done

Output

Anushka
Sakshi
Mahima
Shubham

(E)

```
Anushka Sakshi Mahima Shubham
for a in $@ do
echo $a
```

(E)

Anushka Sakshi Mahima Shubham
for a in \$@ do
echo \$a

(E)

```
Anushka Sakshi Mahima Shubham
for a in $@ do
echo $a
```

(E)

```
Anushka Sakshi Mahima Shubham
for a in $@ do
echo $a
```

(E)

```
Anushka Sakshi Mahima Shubham
for a in $@ do
echo $a
```

(E)

```
Anushka Sakshi Mahima Shubham
for a in $@ do
echo $a
```

(E)

Anushka Sakshi Mahima Shubham
for a in \$@ do
echo \$a

(E)

Expt. No. 10

Page No. 25

Date

Write a shell script to print values from 1 - 10.

```
#!/bin/bash
for s in {1..10}
do
    echo $s
done
```

Output

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

Teacher's Signature :

Write a shell script to find factorial.

```
#!/bin/bash  
read -p 'enter no.' a  
b=1  
while (( a>0 ))  
do  
echo $(( b=b*a ))  
(( a=a-1 ))  
done
```

enter no. 5

120

Output

Write a shell script to print reverse order from 10 - 1

```
#!/bin/bash
n=10
while (( n >= 1 ))
do
echo $n
(( n=n-1 ))
done
```

Output

10	((n = 10))
9	((n = 9))
8	((n = 8))
7	((n = 7))
6	((n = 6))
5	((n = 5))
4	((n = 4))
3	((n = 3))
2	((n = 2))
1	((n = 1))

Write a shell script for until loop.

```
#!/bin/bash  
n=1  
until (( $n>10 ))  
do  
    echo $n  
    (( n++ ))  
done
```

Output

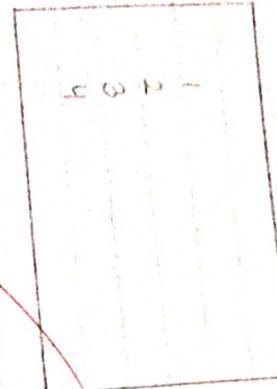
10

Write a shell script of break statement.

```

#!/bin/bash
for i in {1..10}
do
((i>5))
break
done
echo $i

```



Write a shell script of continue statement

```
#!/bin/bash
for i in {1..10}
do
if (( $i==3 )) || (( $i==6 ))
then
done
```

continue

```
fi
else $i
done
```

~~i=0
done~~

Output

1
2
4
5
7
8
9
10

1	2	4	5	7	8	9	10
---	---	---	---	---	---	---	----

Write a shell script to create and print an array.

```
#!/bin/bash
echo "Enter elements for array"
read -a arr
```

```
for ele in ${arr[@]}
```

```
do
```

```
echo $ele
```

```
done
```

~~Output~~ ((3=1 \$1)) ((5=5))

1
2
3
4
5

1
2
3
4
5

1
2
3
4
5

Enter elements for array

Write a shell script to create function.

```
#!/bin/bash
```

```
function show()
```

```
echo $1
```

```
show Anubika
```

```
show Sakshi
```

Anubika
Sakshi

Output

```
[root] ~ % $ ./anshu
Anubika
Sakshi
```