

Host → Esc → Right Control Key

Host + F → Full screen.

Host + Home → Access VM Navigation Bar

Host + Q → Close VM

Host + C → Scale screen mode.

System Call - is a mechanism to access the operating system services.

man write - to see details of the command in man

man 2 write -

q - quit

clear - To clear the screen.

editor ↪ Name

nano w.c → we use c program inside editor

#include <unistd.h>

int main()

```
{
    write(1, "Hello", 5);
}
```

Ctrl + u → For save.

Y → Save modified.

Enter.

gcc w.c → Compile + name of the Program

./a.out → ./ → Current directory  
a.out → default output

Then it will show output:

→ nano w.c

#include <unistd.h>

#include <stdio.h>

int main()

{ int n;

n = write(1, "Hello\n", 6);

printf("Value of n is %d\n", n); }

Fixed File descriptor

Keyboard - 0 - standard input

Screen - 1 - standard output

Screen - 2 - standard error.

File Descriptor ↪

write(1, "Hello", 5);

(What you want to write)

↳ (Count of Alphabets that you show on screen)

Some executable.

#!/bin/sh  
bar  
bar  
Ksh

READ

nano x.c

#include &lt;unistd.h&gt;

int main()

{ char b[30]; → Declares the array

read(0, b, 10); -

write(1, b, 10);

→ All bytes of memory are null integers

#include &lt;unistd.h&gt;

int main() → null values in it - null

{ int n;

char b[30]

n = read(0, b, 30);

write(1, b, n);

→ null

get a file

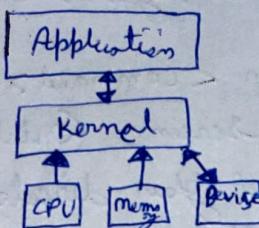
/a.out

make

null

## Command line

**Kernel →**



- Linux is a Kernel.

- Ubuntu is Linux.

distribut

**Shell →** A shell is a computer interface to an operating system.  
Shell exposes the OS's services to human users or other programs.

Shell takes our commands and gives them to operating system to perform.

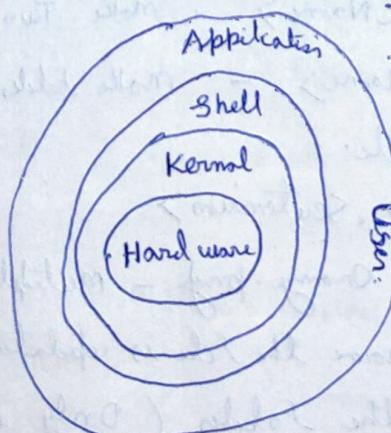
It is named shell because it is outer layer of O.S.

**Terminal →** A terminal is a program that runs a shell.  
Originally, terminals were physical devices but these days we work with software terminals.

Terminal → Shell → O.S

- Bourne shell - (sh)
- GNU Bourne-Again shell - (bash)
- C shell - (csh)
- Korn shell - (ksh)
- Z shell - (zsh)

Various  
Linux-based  
system default  
shell



<innermost> above

Connectable <innermost> above

1. whoami - It show by which name we log in the system.
2. man - To get manual man <command> (q - to quit)
3. clear - To clear the terminal screen (ctrl + L)
4. pwd - Print the current name of your directory.
5. ls - To see the list inside the folder or directory, or parent
  - ls <File name> To check the list inside the file.
  - ls <Folder name>/<Folder name> All
  - ls -l** - Detail with time
6. cd - change the Parent working directory
  - cd <File/Folder name>
  - cd <Folder name>/<Folder Name>
    - For go back use (.) dot.
    - cd .. /
    - Parent Folder.
7. (1) - Root directory (Top level directory)
8. (~) - Home directory
9. mkdir - It helps in make a folder.
  - mkdir <Name>
  - mkdir <Name> <Name> - Make Two Folder
  - mkdir <Name>/<Name> - Make folder inside folder
10. touch - It helps in make file.
  - touch <File Name>, <File Name>
  - ex - touch apple.pdf Orange.png - multiple file  
if u repeat file name the file is updated.
11. rmdir - It helps in delete the folder (Only empty folder will be deleted)
  - rmdir <filename>
  - rmdir <Folder name>/<Folder Name> (Multiple delete)

-v) use this for notification

-r) for move/copy - every file inside folder -

12. rm - It helps in remove File's and or all

rm <filename> (single)

rm <filename> <filename> (multiple remove)

rm -v <filename> (show notifications) (single and empty)

rm -r <filename> (Folders inside folder, nested Folders)

rm -ri <filename> (It ask Confirmation before every single time)

rm -rv <filename> (Notification with nested Folders)

13. Open - Open the File or directory outside the editor.

open <filename>

then he took click on

xdg-open <filename>

14. mv - It help in rename the File or Folder / or move also

mv <old filename> <New filename> - It Rename File in same Path

mv <filename> <Folder name>/ - It help in move the File into Folder.

mv <filename> <filename> <Folder name>/ - Multiple move

mv <Folder name> <Folder name> - Rename Folder

mv <Folder name>/ ..<Folder name> - Move from one folder to another

15. cp - It help in copy the file.

cp <filename> <New filename> - use for same folder/directory make a copy of it.

cp -r <Folder name> <New Folder name> - If ur Folder is not blank then use -r for copy each file and every file in same dir

Copy file to a different Folder with new name

other <file>

16. head - It print the 10 line of the file (default)

head <filename> -> Show Top 10 line of file

head <filename> -n 100 -> Show Top 100 line of file

17. tail - It print bottom 10 line of song.

tail <filename> -> Show bottom 10 line of file.

tail <filename> -n 100 -> Show bottom 100 line of file.

18. date - It show the date.

date > <filename> -> It replace the File content with date.

(>) - override

- If the file don't exist it will make the particular file.

date >> <filename> -> It will add the date output with

(>>) - append file

19. CAT - It give the output of a file.

cat ... <filename>

cat <filename> <filename>

cat <filename> <filename> > <filename>

cat -n <filename> to add line number in file. It make 2 file output in one file.

20. less -

less <filename> It print entire text in terminal

Space - up > b - down

Arrow key.

21. echo -

It print the output argument pass to it print it back.

echo "Hello"

Result -> Hello

It add the text inside the file.

echo "Hello" > <filename> -

echo "I am great" >> <filename> - It add one more line.

22. **wc** - it show 3 parameters No of line, No of words and No of Bytes and Name of File

**wc <filename>** - To show all three parameters in file.

**wc -l <filename>** - No of lines in the file

**wc -w <filename>** - No of words in the file

New method.

{ Pipeline} **ls -l | wc** - current directory list show 3 parameters

**ls -al | wc** - current directory with hidden files.

**date | wc - <filename>**

**cat <filename> <filename> | wc -l > <filename>** - the output of

file is directed in one file with line

23. **sort** - it sort the files (Alphabetically) default of count.

**Sort <filename>**

**sort <filename> > <filename>** - sorted file order is directed

**sort -n <filename>** - For numerical sort in new directory.

**sort -nr <filename>** - For reverse numerical sort.

**sort -nu <filename>** - For numerical with unique no double element present.

24. **uniq** - it will only check only one line above and below by default.

**sort <filename> | uniq -c**  
(It provide the count of item)

(A)  
2 All  
2 Hello  
2 Hi  
2 OF  
1 You  
2 you

**Unique <filename>**

(A)  
Hello  
All  
All  
OF  
You  
OF  
Hi  
Hi  
Hello  
You

(A)  
All  
Hello  
Hi  
OF  
You

**sort <filename> | uniq -c | sort -n**  
(Count will be sorted)

(A)  
1 You  
2 All  
2 Hello  
2 Hi  
2 OF  
2 You

**Sort <filename> | uniq -d**  
(Sort and only print double)

(A)  
All  
Hello  
Hi  
OF  
You

(A) Not available

**sort <filename> | uniq -u** → (A)  
(Sort and only unique value)

remove both double.

## 2.5 Expansions

echo ~ - short cut for home

echo '~' → ~

echo \$PATH →

echo \$USER → It provide Username (environment variable for current user name)

echo \* → It print list of every path name.

echo \*.<extension> → eg - echo \*.txt (It print path name of only that extension file.)

ls -l \*.txt →

ls -l \*.<File extension> → It show long listing information about particular File extension. eg - txt file

echo \*.\*?? → It will find only 2 char matching file extension eg:  
Multiple file of home.py, loga.js etc.

echo \*.\*?? → It will find Only 3 char file extension eg-

rm \*.\*?? → Multiple file like , home.jpg, loga.txt, etc.

rm \*.\*?? → Remove all file having 2 characters file extension.

echo {a,b,c} → It will give 3 char a, b - c

echo {a\*,b\*,c\*}.txt → It will give output a.txt, b.txt, c.txt  
if u use touch it will create 3 separate file with extension.

touch app.{txt,html,py,css} → it will create app.txt, app.html  
& app.py and app.css.

ls app\*

ls <filename>\* → it will show all file which start with app.  
you can also use any character for reaching anything like first alphabet also.

echo {1..99}

→ It will print 1 to 99 in terminal,  
eg - 1, 2 3 4

echo day{1..365}

→ It will print day1 day2 days to day365.

Note - Here we use echo only for show what terminal is doing as output by we can use touch, remove, mkdir, etc instead of echo.

ls -ltr - list you all files according to the order of time in which they created.

vi - It is the text editor

Vim-

example - vim Filenamex.sh

```
#!/bin/bash
echo "I am file"
```

:q!/:wq

cat Filenamex.sh

For execution of any file in linux we command ./ Filenamex.sh.

" " of th file we command sh Filenamex.sh.

**Result - Permission denied**

chmod is use for permission

chmod 777 Filenamex.sh

- USER/owner

./Filenamex.sh

- GROUP

i am file.

- EVERY ONE

**History** - To show what command previously used. (in this machine)

**top** - we ls. show active linux process.

7	4	2	1
↓	↓	↓	↓
READ	WRITE	Execute	

**nproc** - Give the no of CPU

**free** - information of memory of machine (RAM)

**df** - Give the information of all available space in the machine

## Advanced Shellscript

grep - is used to find the matching patterns in file.

Eg -

- vim test.sh

test.sh

```
echo 1
echo 11
echo 12
echo 55
echo 99 -
```

- chmod 777 test.sh

- ./test.sh

Result - 1, 11, 12, 55, 99

- ./test.sh | grep 1

Result - 1, 11, 12

ps -ef - this command is used to show all the currently running processes

Eg with grep -

this is pipe command where to take result to first command and send it to second command.

ps -ef | grep "root" — it will show only that process that start with root.

\* interview - date | echo "today is "

Result - today is (it will not show date because date is kind of std in)

it is belong from std out channel it will receive channel

awk - it help in filters out the statement from output.

The difference between grep and awk is grep show the whole/entire statement but awk is powerful enough to show only particular column of the statement.

Eg - it will show only id (Because 2nd column is id)

- ps -ef | grep root | awk -F " " '{print \$2}'

use case of grep in finding error

eg -

- cat logfile | grep error.
- curl - . is used to retrieve the information from internet  
if logfile exist in servers / cloud ~~we~~ we have to use ~~cat~~ curl with grep.
- wget - it is one of alternative of curl command but it downloads the file where curl reads data without downloading.
- find - use to find the file in which we don't know the location.  
eg -  

```
sudo find / -name prince
      ↓           ↓           ↓
Admin    All      having
          name
      ↓
      any signal
```
- strob - it is use to stop any signal

Shell script for health of VM.

Vim madehealth.sh

```
#####
## Author - Bines
## Date - 01/01/2022
## This script checks Health
## Version - 1
#####
#!/bin/bash
# Set - exec -#
df -h
free -g
mpore
```

→ For run in text mode, if get any error it exit.

- chmod 777 ./madehealth.sh (For basically permission)
- ./madehealth - (For executing the file)