

Expt. No.	Date	Title of the Experiment	Page No.	Date of Submission	Signature of course coordinator
01.	16/1/24	OS installation and post installation task	1-4		8
02.	23/1/24	Virtual box setup & OS configuration	5-7		8
03.	6/2/24	File & directory commands	8-15		8
04.	9/2/24	Commands related to process creation & management	16-21		8
05.	16/2/24	Process Synchronization cmds	22-23		8
06.	20/2/24	Memory Management commands	24-25		8
07.	1/3/24	Shell Programming	26-29		8
08.	4/3/24	Automation of System Task	30-33		8
09.	7/3/24	Network Management Cmds	34-36		8
10.	7/4/24	User Authentication Cmds	37-40		8
11.	12/4/24	System Monitoring & Setup	41-43		8
12.	20/4/24	FTP & Apache Server Setup	44-46		8
13.	27/4/24	Storage Management Cmds.	47-48		8

COURSE OBJECTIVES:

- c01: Explain functions and services of an operating system
- c02: Create a virtual environment and configure it to meet a specific application requirement.
- c03: Identify & use Linux commands to create & manage simple file processing operations, organize directory structure and develop shell script to automate given simple task.
- c04: Demonstrate the role and responsibilities of a Linux system administrator and analyse problems using suitable diagnostic tools and resolve issues.

WEEK-01

* OS DEFINITION

An Operating System (OS) is system software that manages computer hardware, software resources and provides common services for computer programs. In other words, an Operating System (OS) is software that acts as an interface between computer hardware components and the user.

* TYPES OF INSTALLATION

• ATTENDED INSTALLATION:

→ An installation process usually needs a user who attends it to make choices, such as accepting or declining an end-user license agreement (EULA), specifying preferences such as the installation location, supplying passwords or assisting in product activation.

→ In graphical environments, installers that offer a wizard-based interface are common.

- UNATTENDED INSTALLATION

→ An installation that is performed without user interaction during its progress or with no user present at all.

→ One of the reason to use this approach is to automate the installation of large number of system.

- HEADLESS INSTALLATION

→ Installation performed without using a computer monitor connected.

→ In unattended forms of headless installation, another machine connects to the target machine connects to the target (for example, via a local network) and takes over the display output.

- SCHEDULED OR AUTOMATED INSTALLATION

→ An installation process that runs on a preset time or when a predefined condition meet the requirements, as opposed to an installation process that starts explicitly on a user's command.

→ An operating system may automatically install a device driver for a device that the user connects.

* BOOT METHODS

Booting is a process of starting a computer as initiated via hardware such as a button or by a software command.

TYPES OF BOOTING

→ WARM BOOTING :- The warm booting is that in which system state from the starting or from initial state means.

In the warm booting the system will be started from its beginning state means, first, the user will press the power button, then this will read all the instruction from the ROM and the Operating System will be automatically gets loaded into the system (RAM).

→ COLD BOOTING :
The cold booting is that in which system automatically starts when the system is in running state.

- For example, due to light fluctuation, the system will automatically restart. In this, chances of damaging of system are more. The system will now be start from its initial state so some files may be damaged because they are not properly stored into the system.

* FILE SYSTEM AND FORMATTING

• FILE SYSTEM:-

→ A file system is a process of managing how and where data is stored on a storage disk, which is also referred to as file management or FS.

→ It is a logical disks component that compresses files separated into groups, which is known as directories.

→ The file system enables user to view a file in the current directory as files are often managed in a hierarchy.

→ It is abstract to a human user and related to computer's hence it manages a disk's internal operations.

- NTFS is the most common file system in modern times (Windows OS)
- Without file management it would be impossible for a file with the same name to exist and also impossible to remove installed programs and recover specific files.

EXAMPLES OF FILE SYSTEMS:-

- FAT (File Allocation Table)
 - FAT 8
 - FAT 12
 - FAT 32
 - FAT 16
- GFS (Global file system)
- HFS (Hierarchical file system)
- NTFS (New Technology file system)
- UDF (Universal Disk format)

* FORMATTING :-

- Is a process of preparing the storage device to store the data.
- Formatting storage device will erase the earlier contents of the devices.

* POST INSTALLATION TASKS :-

Post installation task is the set of steps to be carried out to ensure that the installation is complete and went smoothly.

POST INSTALLATION TASKS FOR UBUNTU OPERATING SYSTEM :-

• ONLINE Accounts :-

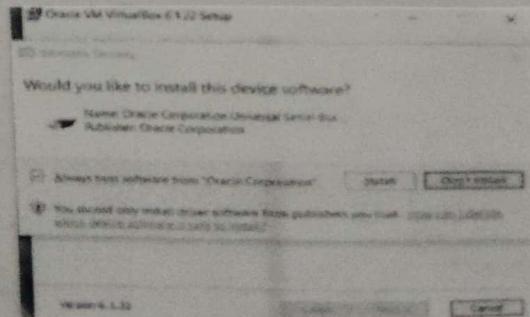
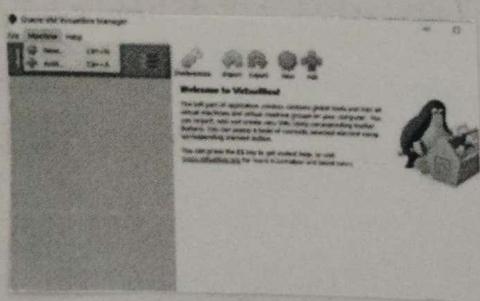
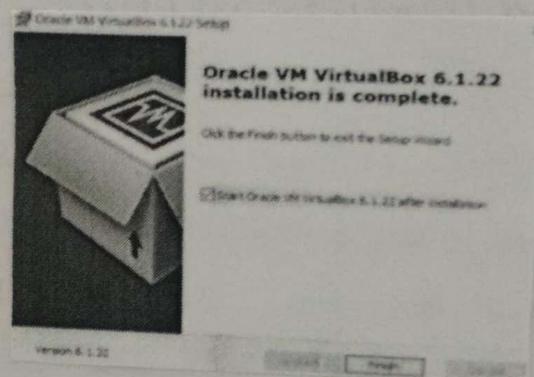
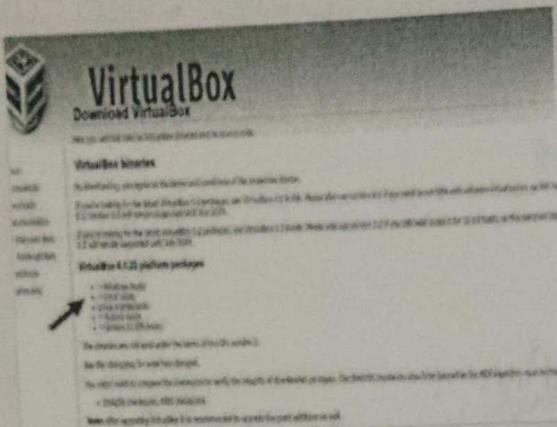
The first step allows user to configure online accounts, in case we want to integrate the desktop with different services.

• LIVE PATCH :- is a service that allows the installation of some updates that would generally require a system reboot, such as those of the kernel.

• HELP IMPROVE UBUNTU :- In this step user can choose whether or not to send data from his system to Ubuntu. The option is activated by default, the user can verify the secrecy of the data being sent beforehand. The results are used to improve Ubuntu.

• PRIVACY :- If required, a user can enable location services so that apps can determine user geographic location. All the application installed by default on Ubuntu are free software.

• YOU ARE READY TO START.



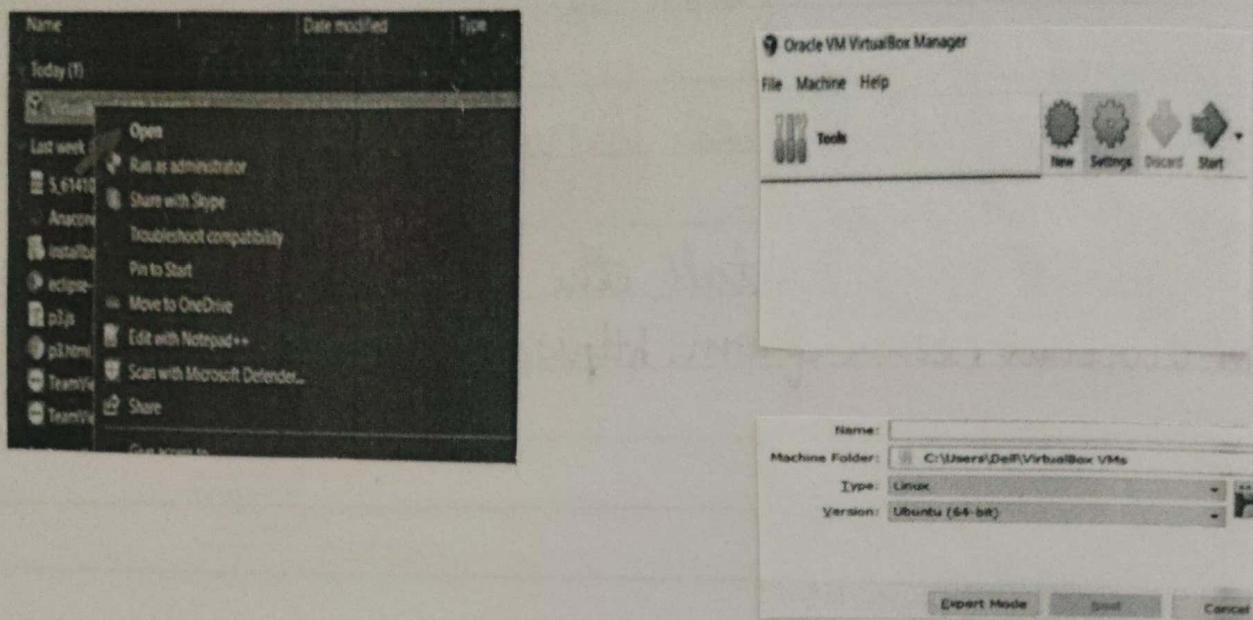
WEEK-D2

INSTALL AND CONFIGURE VIRTUAL MACHINE - VIRTUAL BOX.

- Download and install the latest version of Oracle VM Virtual Box (from: <https://www.virtualbox.org/wiki/Download>)

01. INSTALLATION STEPS.

- Double click or right click on the Oracle VM Virtual Box executable file.
- "Setup Wizard" screen appears click next.



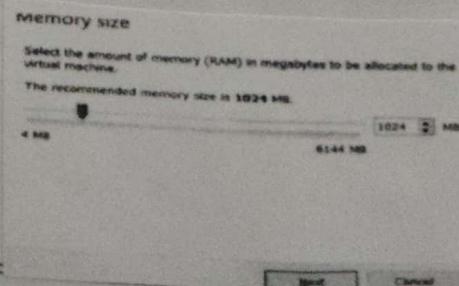
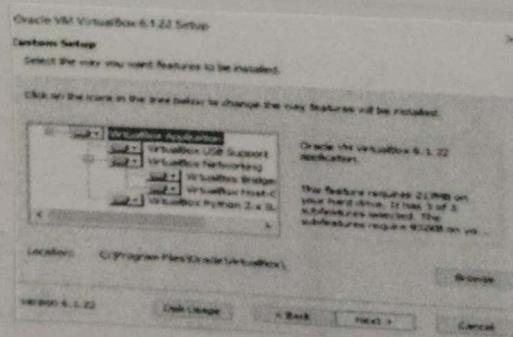
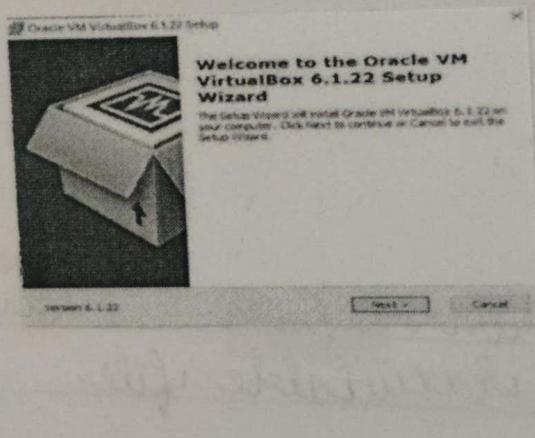
Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icons.

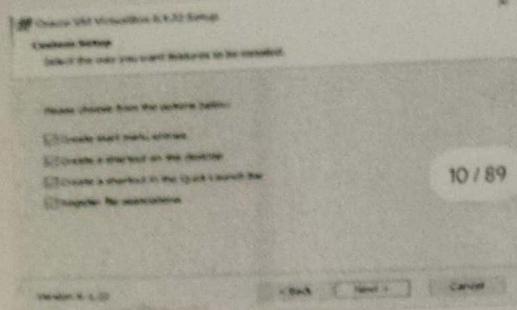
If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is 10.00 GB.

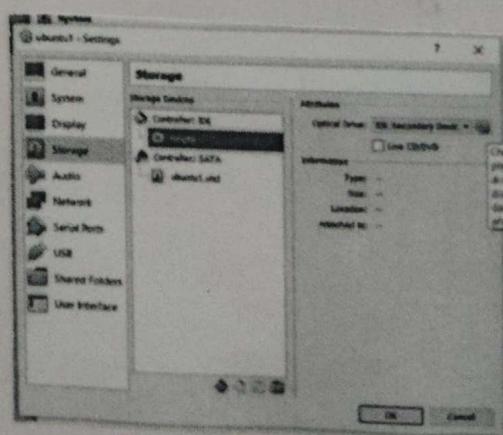
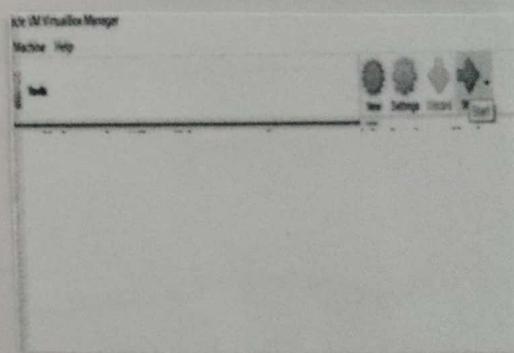
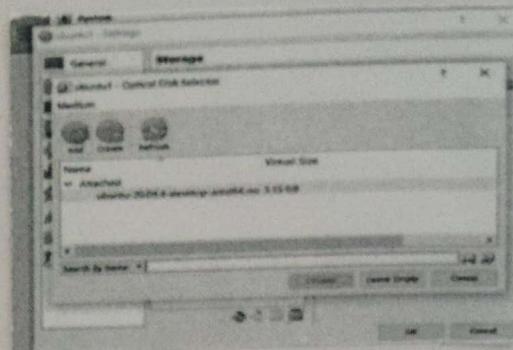
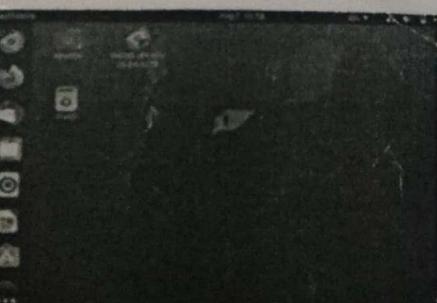
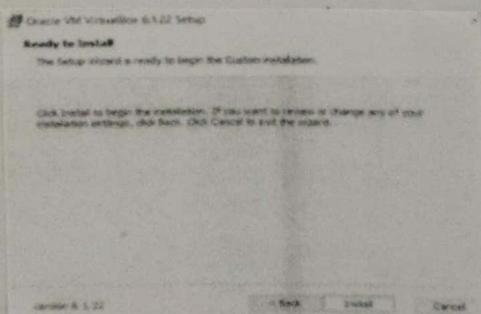
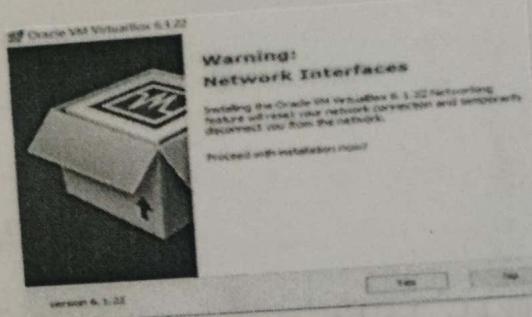
- Do not add a virtual hard disk.
- Create a virtual hard disk now.
- Use an existing virtual hard disk file.



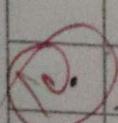
- Then "Custom setup with all the features of WMP" screen appears click next.
- Next "Select the way you want features to be installed" Screen appears click next.
- Next "Warning network interface" Screen appears click yes.



Version 6.1.22 Back Next > Cancel



- And last "Ready to install" screen appears click on install button.
- While installing another dialogue box appears asking "Would you like to install this device software?" click on install.
- After completion of installation click on the finish button.



So that installation of virtual box completes you can use it.

Ques:

Activities Terminal

kiyotoka@kiyotoka-VirtualBox:~\$ ls
To run a command as administrator (use "sudo"), use the "Run as root" menu item.
See "man sudo_root" for details.
Desktop Documents Downloads Music Pictures Public Templates Videos
Desktop 2 kiyotoka kiyotoka 0906 Feb 8 10:17 Desktop
Desktop 2 kiyotoka kiyotoka 0906 Feb 8 10:17 Documents
Desktop 2 kiyotoka kiyotoka 0906 Feb 8 10:17 Downloads
Desktop 2 kiyotoka kiyotoka 0906 Feb 8 10:17 Music
Desktop 2 kiyotoka kiyotoka 0906 Feb 8 10:17 Pictures
Desktop 2 kiyotoka kiyotoka 0906 Feb 8 10:17 Public
Desktop 2 kiyotoka kiyotoka 0906 Feb 8 10:17 Templates
Desktop 2 kiyotoka kiyotoka 0906 Feb 8 10:17 Videos

Activities Terminal

kiyotoka@kiyotoka-VirtualBox:~\$ pwd
/home/kiyotoka
kiyotoka@kiyotoka-VirtualBox:~\$

Activities Terminal

Feb 8 12:09

kiyotoka@kiyotoka-VirtualBox:~\$ mkdir sjp
kiyotoka@kiyotoka-VirtualBox:~\$ ls
Desktop Documents Downloads Music Pictures Public sjp Templates Videos
kiyotoka@kiyotoka-VirtualBox:~\$

WEEK-03

FILE AND DIRECTORY COMMANDS.

ls (list)

- * Use ls without any arguments to display directory content.
- * ls - a or begin with a "dot" indicates hidden file.
- * ls - l long listing of file.

Eg:- ~~-rwxr-xr-x 1 Smith Staff 10030 Mar 23 10.41 sm.txt~~

1 2 3 4 5 6 7

1 = access modes/permission

2 = no. of links

3 = Owner

4 = igroup

5 = Size

6 = Date

7 = Name of file

pwd (Present working directory)

- * will give present working directory path info

Syntax: pwd

mkdir (Make directory)

mkdir will create a new directory (folder).

Syntax: mkdir directoryname

Eg: mkdir sjp.

```
kiyotoka@kiyotoka-VirtualBox:~$ cd sjp  
kiyotoka@kiyotoka-VirtualBox:~/sjp$ touch manasa.txt  
kiyotoka@kiyotoka-VirtualBox:~/sjp$ vi manasa.txt  
hello world  
kiyotoka@kiyotoka-VirtualBox:~/sjp$ 
```

```
Terminal ▾ Feb 8 12:09  
kiyotoka@kiyotoka-VirtualBo  
gkiyotoka-VirtualBox:~$ mkdir sjp  
@kiyotoka-VirtualBox:~$ ls  
Documents Downloads Music Pictures Public sjp Templates Videos  
gkiyotoka-VirtualBox:~$ cd sjp  
gkiyotoka-VirtualBox:~/sjp$ cd..  
command not found  
@kiyotoka-VirtualBox:~/sjp$ cd ..  
@kiyotoka-VirtualBox:~$ rmdir sjp  
@kiyotoka-VirtualBox:~$ ls  
Documents Downloads Music Pictures Public Templates Videos  
@kiyotoka-VirtualBox:~$ 
```

```
Activities Terminal ▾  
hello world
```

`cd` (change directory)

`cd` is used to change directory.

* Change to home directory `cd`

`cd ~`

* Go up one level back to the current directory `cd ..`

`rmdir` (Remove directory)

* Will remove the directory [Selected directory]

* Make sure the folder is empty & current directory is not part of directory being deleted.

Syntax: `rmdir dirname`

Example: `rmdir college`

→ Creating files using vi Editor:

* Command mode.

all the keys works as command. These keys are used for inserting, appending, deleting, opening new line.

* Insert mode

User can insert, append, edit or replace text. We can switch from the command mode to insert mode by pressing `Esc + i`

* Esc command mode.

Used to enter command at the bottom of line in vi editor called command line.

Activities

Terminal ▾



```
kiyotoka@kiyotoka-VirtualBox:~$ cd sjp
kiyotoka@kiyotoka-VirtualBox:~/sjp$ touch manasa.txt
kiyotoka@kiyotoka-VirtualBox:~/sjp$ vi manasa.txt
```

vi editor saving and quitting command:

w: Save the contents of the file.

q: Quit from vi editor

wq: Save and quit from vi editor.

To create a sample file in editor open terminal & type.

* Syntax: vi filename

* Example: vi es.txt

To get into insert mode from command mode type Esc + i

To get into execution mode from insertion mode press Esc + Shift + :

→ File Manipulation commands

a) Creating file.

"touch" command will create empty file

* Syntax: touch filename

* Example: touch es.txt.

b) Display contents of file

* Syntax: cat filename

* Example: cat es.txt.

Activities Terminal kiyot

```
yotoka@kiyotoka-VirtualBox:~/sjp$ touch shiva.txt
yotoka@kiyotoka-VirtualBox:~/sjp$ cp manasa.txt shiva.txt
yotoka@kiyotoka-VirtualBox:~/sjp$ cat shiva.txt
Hello world
yotoka@kiyotoka-VirtualBox:~/sjp$ 
```

```
kiyotoka@kiyotoka-VirtualBox:~/sjp$ cat manasa.txt
Hello world
kiyotoka@kiyotoka-VirtualBox:~/sjp$ rm shiva .txt
rm: cannot remove 'shiva': No such file or directory
rm: cannot remove '.txt': No such file or directory
kiyotoka@kiyotoka-VirtualBox:~/sjp$ ls
manasa.txt shiva.txt
kiyotoka@kiyotoka-VirtualBox:~/sjp$ rm shiva.txt
kiyotoka@kiyotoka-VirtualBox:~/sjp$ l
manasa.txt
kiyotoka@kiyotoka-VirtualBox:~/sjp$ 
```

Activities Terminal Feb 8 12:09 kiyot

```
ka@kiyotoka-VirtualBox:~$ mkdir sjp
ka@kiyotoka-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public sjp Templates Videos
ka@kiyotoka-VirtualBox:~$ 
```

c) ~~removefile~~

- * Syntax: rm filename
- * Example: rm eee.txt

d) copy file (cp)

* Used to create a copy of a existing file.

- * Syntax: cp file1 file2
- * cp /path/file1 /path/file2

* Example: cp cs.txt nba.txt

* New file of nba.txt is created & content of nba.txt is same as cs.txt

e) Move [mv]

* Used to rename the file

* Syntax: mv oldfilename newfilename

* Example: mv cs.txt comsci.txt

→ Pipes: The pipe command lets user sends the output of one command to another. Piping is the term suggest can redirect the standard output, input or error of one process to another for future processing.

~~Syntax: cat filename | sort~~

~~Example: cat ec.txt | sort~~

Activities DJ Terminal ~

ubuntu@ubuntu: ~

```
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures  Templates  headtail.txt  knew.c
Documents  Music  Public  Videos    kk.txt  polytechnic
ubuntu@ubuntu:~$ cat headtail.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
line11
line12
line13
ubuntu@ubuntu:~$ head headtail.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
```

f) more:

It's a filter for paging through text one screenful at a time.

- * Syntax: more filename

- * Press spacebar/Return key to read remaining parts of file.
- * Press q to quit.

g) less:

It is used to viewing the file instead of opening the file. Similar to "more" cmd but allows backward or forward movement.

- * Syntax: less filename.

h) cmp [compare]:

Compare two files byte to byte.

- * Syntax: cmp file1 file2

- * Example: cmp md.txt mansa.txt

i) Head & tail

- * head - by default outputs first 10 lines of file.

- * Syntax: head filename

~~head -5 filename [displays first 5 lines]~~

- * tail - by default outputs last 10 lines of file.

- * Syntax: ~~tail~~ filename

~~tail -3 filename [displays last 3 lines]~~

- * Example :- head -2 consci.txt

~~tail -3 ec.txt~~

Activities Terminal kiyotoka@kiyotoka-VirtualBox:~\$ cd sjp
kiyotoka@kiyotoka-VirtualBox:~/sjp\$ ls -l
total 8
-rw-rw-r-- 1 kiyotoka kiyotoka 13 Feb 8 12:11 manasa.txt
-rw-rw-r-- 1 kiyotoka kiyotoka 33 Feb 8 12:16 raj.txt
kiyotoka@kiyotoka-VirtualBox:~/sjp\$ chmod 777 raj.txt
kiyotoka@kiyotoka-VirtualBox:~/sjp\$ ls -l
total 8
-rw-rw-r-- 1 kiyotoka kiyotoka 13 Feb 8 12:11 manasa.txt
-rwxrwxrwx 1 kiyotoka kiyotoka 33 Feb 8 12:16 raj.txt
kiyotoka@kiyotoka-VirtualBox:~/sjp\$

Activities Terminal kiyotoka@kiyotoka-VirtualBox:~\$ cd sjp
kiyotoka@kiyotoka-VirtualBox:~/sjp\$ cat raj.txt
Feb 2024
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
kiyotoka@kiyotoka-VirtualBox:~/sjp\$

File Permission

- * There are three users in linux namely owner, group (g), others (o).
- * For all user can be identified with letters.
- * All 3 user will have 3 possible permission namely read (r), write (w), execute (e).
- * number assigned for permission read(4), write(2), execute(1).
- * Command used to change permission of file chmod.
- * Syntax: chmod filepermissionpattern filename.

Working with numbers in chmod.

- * Syntax: chmod chmodnumber filename.

Ig: If user wants rwx rwx - format is set number
to 765 - rwx rwx - x

654 - rw- r- x - -

→ File compression & decompression

~~Compression reduces the size of an application or document for storage or transmission. compressed file are smaller, download fast.~~

Activities Terminal Feb 8 12:23
kiyotoka@kiyotoka-VirtualBox:~/sjp\$

```
kiyotokagkiyotoka-VirtualBox:~/sjp$ gzip raj.txt
kiyotokagkiyotoka-VirtualBox:~/sjp$ cat raj.txt.gz
you
hello world
world
we
word
you
kiyotokagkiyotoka-VirtualBox:~/sjp$ gunzip raj.txt.gz
kiyotokagkiyotoka-VirtualBox:~/sjp$ cat raj.txt
hello world
world
we
word
you
kiyotokagkiyotoka-VirtualBox:~/sjp$
```

you
kiyotoka@kiyotoka-VirtualBox:~/sjp\$ sort raj.txt
hello world
i
we
word
world
you
kiyotokagkiyotoka-VirtualBox:~/sjp\$

world
you
kiyotoka@kiyotoka-VirtualBox:~/sjp\$ uniq manasa.txt
hello world
kiyotoka@kiyotoka-VirtualBox:~/sjp\$

- * Compresses the file & a file with .gz extension
- * Syntax - gzip filename
cat filename.gz
- * Example - gzip md.txt
cat md.txt.gz.

Decompressing file

To decompress we write gunzip
Syntax: gunzip filename with .gz

Eg:- gunzip md.

Text processing commands

Commands affecting test & test files

→ Sort → used to sort a file, arranging the order in a particular order.

By default, the sort command sorts file assuming the contents are ASCII

Syntax: sort filename.

Input : sort.

→ uniq: uniq filter removes duplicate lines from a sorted file. It is often seen in a pipe coupled with sort.

Syntax: uniq filename

Input: uniq manasa.txt

kiyotoka@kiyotoka

```
kiyotoka@kiyotoka-VirtualBox:~/sjp$ cut -c 1-2 raj.txt
he
o
o
e
l
yo
kiyotoka@kiyotoka-VirtualBox:~/sjp$ greppatt raj.txt
greppatt: command not found
kiyotoka@kiyotoka-VirtualBox:~/sjp$ grep pattern raj.txt
kiyotoka@kiyotoka-VirtualBox:~/sjp$ cat raj.txt
hello world
world
word
we
L
you
kiyotoka@kiyotoka-VirtualBox:~/sjp$ ]
```

→ **cut**: a tool for extracting fields from files
Important options are -d which specifies delimiter to separate fields -f which indicates field numbers.

Syntax: cut -c 1-2 filename.

→ **grep**: a multiple file search tool that uses regular expression.

Syntax: grep pattern filename.

→ **wc**: this command gives a "word count" on a file or I/O stream

Syntax: wc

8/
8) ✓ (v)

Process creation and management.

- * A process is any active (running) instance of program in other words, process is a program in execution.
- * A new process can be created by fork() sys call.
- * Existing process is called the parent process & the process is created newly is called child process.
- * Each process is given a unique process identification number (PID).

Starting a Process

→ Process execute in two ways:

* Foreground process * Background process.

→ Foreground process

* When a program is running in the foreground user can't start another process.

→ Background Process

* To enable the background process, provide an ampersand symbol at the end of command. They run in background & usually do not need user input for example Antivirus.

Activities Terminal ▾

Feb 8 12:27
kiyotaka@kiyotaka-VirtualBox:~/sjp\$ sleep --help

Usage: sleep NUMBER[SUFFIX]...
or: sleep OPTION

Pause for NUMBER seconds. SUFFIX may be 's' for seconds (the default), 'm' for minutes, 'h' for hours or 'd' for days. Unlike most implementations that require NUMBER be an integer, here NUMBER may be an arbitrary floating point number. Given two or more arguments, pause for the amount of time specified by the sum of their values.

--help display this help and exit
--version output version information and exit

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>
Full documentation at: <<https://www.gnu.org/software/coreutils/sleep>>
or available locally via: Info '(coreutils) sleep invocation'

kiyotaka@kiyotaka-VirtualBox:~/sjp\$

Activities Terminal ▾

kiyotaka@kiyotaka-VirtualBox:~/sjp\$ sleep 13&

[2] 2819

kiyotaka@kiyotaka-VirtualBox:~/sjp\$ exec >raj.txt

[1]- Done sleep 13

[2]+ Done sleep 13

kiyotaka@kiyotaka-VirtualBox:~/sjp\$ cal

kiyotaka@kiyotaka-VirtualBox:~/sjp\$ exit

→ sleep: This command is used to create a dummy job, helps in delaying the execution.

Syntax: sleep numbers (suffix).

→ jobs: The jobs command will list all jobs on the system: active, stopped or otherwise.

Syntax: jobs

Example: [1] stopped sleep 500
 ↓
 Job id

→ ps: lists the processes running in current terminal

Syntax: ps

→ Demonstrate exec() system call

The exec() family of functions replaces the current processing image with new process image. exec command in linux is used to execute a command from the bash itself.

Syntax: exec > filename

Input: exec > out.txt

val

exit

cat out.txt

Activities Terminal Feb 8 12:3
kyotokagkiyotoka-Vir

```
kyotokagkiyotoka-VirtualBox:~/s]$ bg sleep 2s
[1] 2859
kyotokagkiyotoka-VirtualBox:~/s]$ bash: bg: no job control
[1]+ Exit 1                  bg sleep 2s
kyotokagkiyotoka-VirtualBox:~/s]$ jobs
kyotokagkiyotoka-VirtualBox:~/s]$ bg%1
bg%1: command not found
kyotokagkiyotoka-VirtualBox:~/s]$ bg sleep 2000
bash: bg: sleep: no such job
bash: bg: 2000: no such job
kyotokagkiyotoka-VirtualBox:~/s]$ bg sleep 2000
[1] 2871
kyotokagkiyotoka-VirtualBox:~/s]$ bash: bg: no job control
jobs
[1]+ Exit 1                  bg sleep 2000
kyotokagkiyotoka-VirtualBox:~/s]$ bg%1
bg%1: command not found
kyotokagkiyotoka-VirtualBox:~/s]$ bg %1
bash: bg: %1: no such job
kyotokagkiyotoka-VirtualBox:~/s]$ bg %1
bash: bg: %1: no such job
kyotokagkiyotoka-VirtualBox:~/s]$ jobs
kyotokagkiyotoka-VirtualBox:~/s]$ bg %
bash: bg: %: no such job
```

→ bg : background

bg is used to place foreground jobs in background

Syntax: bg jobnumber.

Eg: Sleep 500 is a command which is used to create dummy job.

\$ jobs

\$ Sleep 500

[1]+ Stopped Sleep 500.

\$ jobs

[1]+ Stopped Sleep 500

\$ bg .1

[1]+ Sleep 500 &

\$ jobs

[1] Running Sleep 500

→ fg - foreground.

~~fg is a command that moves a background process on current linux to the foreground.~~

~~Syntax: fg jobnumber.~~

Eg: \$ Sleep 500

\$ jobs

* fg .1

* Sleep 500 // waiting for process to complete.

Activities

Terminal ▾

kyoto

```
kiyotoka@kiyotoka-VirtualBox:~/sjp$ nohup sort raj.txt  
nohup: ignoring input and appending output to 'nohup.out'  
kiyotoka@kiyotoka-VirtualBox:~/sjp$ cat nohup.out
```

```
11 12 13 14 15 16 17  
     1 2 3  
18 19 20 21 22 23 24  
25 26 27 28 29  
4 5 6 7 8 9 10  
February 2024  
Su Mo Tu We Th Fr Sa
```

```
11 12 13 14 15 16 17  
     1 2 3  
18 19 20 21 22 23 24  
25 26 27 28 29  
4 5 6 7 8 9 10  
February 2024  
Su Mo Tu We Th Fr Sa
```

```
kiyotoka@kiyotoka-VirtualBox:~/sjp$ ]
```

→ nohup

The nohup command is one way of blocking the signup signal & allowing a process to complete even after logging out from the terminal shell.

SIGUP [Signal Hang up] is a signal that terminates a linux processes when its controlling terminal is closed.

Syntax: nohup command arguments.

Input: nohup sort test.txt
→ cat nohup.out

(ii) Stopping a process

→ kill → terminates running processor on a linux machine.

Syntax: -kill PID

PID [Process ID]

Eg: > sleep 500 &

> ps

PID	TTY	Time	end
6369	pts/10	00:00:00	Sleep

> kill 6369

> ps

Activities

Terminal ▾

```
kiyotaka@kiyotaka-VirtualBox:~/sjp$ nice -5 wc -18
[1] 3241
kiyotaka@kiyotaka-VirtualBox:~/sjp$ wc: invalid option -- '1'
Try 'wc --help' for more information.
```

Activities Terminal ▾ FEB 8 12:33 kiyotaka@kiyotaka-VirtualBox: /tmp/crontab.W98D56/crontab

```
GNU nano 4.8
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timestones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
# n h dom mon dow   command
```

Activities Terminal ▾ FEB 8 12:33 kiyotaka@kiyotaka-VirtualBox: /tmp/crontab.W98D56/crontab

```
GNU nano 4.8
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timestones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
# n h dom mon dow   command
```

→ **pkill**: used to send signal to kill process by process name.

Syntax: pkill -processname

Eg: pkill firefox

→ **nice**: helps in execution of a program process with modified scheduling priority.

Syntax: nice option command [args]

→ **top**: this cmd has been around for long time & is very useful for viewing details of running processes & quickly identifying issues such as memory leaks.

Syntax: top

iii Commands to schedule tasks.

→ **cron**.

~~cron can be defined as a software utility that is provided by a linux like OS that operates the scheduled operation on a predetermined period.~~

Syntax: crontab -e

Activities Terminal

Feb 8 12:52
kiyotaka@kiyotaka-VirtualBox:~\$ sudo apt install at
[sudo] password for kiyotaka:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
libfl2
Suggested packages:
default-nta | mail-transport-agent
The following NEW packages will be installed:
at libfl2
Upgraded, 2 newly installed, 0 to remove and 274 not upgraded.
Need to get 58.1 kB of archives.
After this operation, 241 kB of additional disk space will be used.
Do you want to continue? [Y/n] []

kiyotaka@kiyotaka-VirtualBox:~\$ at now
warning: commands will be executed using /bin/sh
st> cal >> manasa.txt
st> <END>
Job 2 at Thu Feb 8 12:56:00 2024
kiyotaka@kiyotaka-VirtualBox:~\$ cat manasa.txt
Hello world
February 2024
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
February 2024
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
kiyotaka@kiyotaka-VirtualBox:~\$ []

→ cat

- * Jobs created with cat command are executed only once.
- * Syntax : cat option runtime.

Working with command.

- * Command to list the users pending jobs
 > cat -l or > atq
- * Schedule a job for the coming monday at a time 20 mins later than the current time.
 > at monday +20 minutes.
- * Schedule a job to run cat 1:45 Aug 12 2020.
 > at 1:45 081220.

8/12/10

✓

```
codebine@DESKTOP-LG8NTJ8:~$ ps -t
 PID TTY      STAT   TIME COMMAND
 134 tty3      Ss    0:00 /init
 135 tty3      S     0:00 -bash
 146 tty3      R     0:00 ps -t
codebine@DESKTOP-LG8NTJ8:~$ -
```

```
codebine@DESKTOP-LG8NTJ8:~$ top
Tasks:  6 total,  1 running,  5 sleeping,  0 stopped,  0 zombie
%Cpu(s):  0.6 us,  1.6 sy,  0.0 ni, 97.4 id,  0.0 wa,  0.4 hi,  0.0 si,  0.0 st
MiB Mem :  8089.1 total,  5286.8 free,  2578.4 used,   224.0 buff/cache
MiB Swap: 20672.8 total, 20634.8 free,      38.0 used.  5380.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+ COMMAND
128	codebine	20	0	15836	2316	1596	R	0.2	0.0	0:00.09 top
1	root	20	0	8980	440	368	S	0.0	0.0	0:00.40 init
72	root	20	0	9320	244	180	S	0.0	0.0	0:00.00 init
73	codebine	20	0	14116	3784	3560	S	0.0	0.0	0:00.13 bash
116	root	20	0	9328	244	188	S	0.0	0.0	0:00.00 init
117	codebine	20	0	14116	3784	3688	S	0.0	0.0	0:00.14 bash

WEEK-05

PROCESS SYNCHRONIZATION

→ Commands to exhibit thread concepts:

- * Threads are a popular programming abstraction for parallel execution on modern operating systems.
- * When threads are forked inside a program for multiple flows of execution, these threads share certain resources.
- * These properties make threads an efficient mechanism for concurrent execution.
- * Each thread will then have its own thread ID (TID).
- * Threads are nothing more than standard processes which happen to share certain resources.

→ Using the ps

- * The "-T" option for the ps command enables thread views.
- * The "SPID" column represents thread IDs, and "CMD" column shows thread names.

→ Using the top

The top command can show a real-time view of individual threads. To enable thread views in the top output, invoke top with "-H" option. This will list all Linux threads.
Ex:- top. To restrict this output to a particular process and check all threads running inside the process.

Ex-2: top -H -p 1572

0[1[2[3[Mem[Swap[10.9%]	Tasks: 6, 1 thr; 1 running	
						3.3%]	Load average: 0.52 0.58 0.59	
						7.3%]	Uptime: 00:04:54	
						3.3%]		
						2.59G/7.90G]		
						22.6M/20.2G]		
PID	USER	PRI	NI	VIRT	RES	SHR	S CPU% MEM%	TIME+ Command
1	root	20	0	8972	432	356	S 0.0 0.0 0.0	0:00.39 /init
8	root	20	0	8972	432	356	S 0.0 0.0 0.0	0:00.00 /init
72	root	20	0	9320	244	180	S 0.0 0.0 0.0	0:00.00 /init
73	codebine	20	0	14116	3784	3560	S 0.0 0.0 0.0	0:00.13 -bash
91	root	20	0	9320	244	184	S 0.0 0.0 0.0	0:00.00 /init
92	codebine	20	0	14116	3784	3684	S 0.0 0.0 0.0	0:00.16 -bash
103	codebine	20	0	13784	2880	1744	R 0.0 0.0 0.0	0:00.14 htop

→ Using htop

- * A more user-friendly way to view threads per process is via htop, an ncurses-based interactive process viewer.
- * This program allows user to monitor individual threads in tree views.
- * To enable thread views in htop, launch htop, & press F2 to enter htop setup menu.
- * Choose "Display option" under "Setup" column, and toggle on "Tree view" and "Show custom thread name" options. Press F10 to exit the setup.

8/12
16
③

```
Ubuntu Terminal * Aug 10 05:56  
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ free  
total used free shared buff/cache available  
Mem: 2705828 772872 500816 238048 1400340 1504240  
Swap: 0 0 0  
ubuntu@ubuntu:~$
```

```
codebine@DESKTOP-LG8NTJ8:~/ $ sudo cat /proc/meminfo  
[sudo] password for codebine:  
MemTotal: 8283224 kB  
MemFree: 5372716 kB  
Buffers: 34032 kB  
Cached: 188576 kB  
SwapCached: 0 kB  
Active: 167556 kB  
Inactive: 157876 kB  
Active(anon): 103104 kB  
Inactive(anon): 17440 kB  
Active(file): 64452 kB  
Inactive(file): 140436 kB  
Unevictable: 0 kB  
Mlocked: 0 kB  
SwapTotal: 21168936 kB  
SwapFree: 21145840 kB  
Dirty: 0 kB  
Writeback: 0 kB  
AnonPages: 102824 kB  
Mapped: 71404 kB  
Shmem: 17720 kB  
Slab: 13868 kB  
SReclaimable: 6744 kB  
SUnreclaim: 7124 kB  
KernelStack: 2848 kB  
PageTables: 2524 kB  
NFS_Unstable: 0 kB  
Bounce: 0 kB  
WritebackTmp: 0 kB  
CommitLimit: 515524 kB  
Committed_AS: 3450064 kB  
VmallocTotal: 122880 kB  
VmallocUsed: 21296 kB  
VmallocChunk: 66044 kB  
HardwareCorrupted: 0 kB  
AnonHugePages: 2048 kB  
HugePages_Total: 0
```

WEEK-06

MEMORY MANAGEMENT

→ free command

Commands to view memory consumption.

→ /proc/meminfo

- * Check memory usage by reading the file at /proc/meminfo.
- * Know that the /proc file system does not contain real files.
- * They are rather virtual files that contain dynamic information about the kernel and the system.
- * Check the values of MemTotal, MemFree, Buffers, Cached, SwapTotal, SwapFree. They indicate same values of memory usage as the free command.

Terminal Aug 10 06:00 en: ~

```
ubuntu@ubuntu:~$ vmstat
procs      memory      swap      io      system      cpu
r b swpd   free   buff   cache   st   so   bt   bo   in   cs us sy id wa s
0 0     0 390288   5104 1401240   0   0   240   0 507 223 2 1 97 0
ubuntu@ubuntu:~$
```

→ `vmstat`

* `vmstat -s`

* Portion of output is

* The top few lines indicate total memory, free memory etc and so on.

→ `top` command

* The `top` command is generally used to check memory and CPU usage per process.

* It also reports total memory usage and can be used to monitor the total RAM usage.

* The header in output has the required information.

* Check the KiB Mem and KiB Swap lines in the header. They indicate total, used and free amounts of the memory. The buffer and cache information is present here too.

*

→ `htop`

* Similar to the `top` command, the `htop` command also shows memory usage along with various other details.

Q) * The header in `top` shows CPU usage along with RAM and swap usage with the corresponding figures.

10

```
vboxuser@hello:~$ gedit  
vboxuser@hello:~$ chmod 777 Biggestofthreenumbers.sh  
vboxuser@hello:~$ ./Biggestofthreenumbers.sh  
Enter three numbers  
33 66 82  
Biggest among 33 66 82 is 82  
vboxuser@hello:~$
```

WEEK-07 SHELL PROGRAMMING,

01. Biggest of Three Numbers using if-elif-else

Find biggest of Three Numbers

echo Enter three numbers.

read a b c

if ((\$a >= \$b && \$a >= \$c))

then

big=\$a

elif ((\$b >= \$a && \$b >= \$c))

then

big=\$b

elif ((\$c >= \$a && \$c >= \$b))

then

big=\$c

fi

echo Biggest among \$a \$b \$c is \$big

```
vboxuser@hello:~$ ./Reverse.sh  
Original String:Helloo  
Reversed String:olleH  
vboxuser@hello:~$ █
```



```
REVERSE STRING : $ ./Age.sh  
enter your age  
17  
you are younger!!  
VOTING ELIGIBILITY: $ ./Age.sh  
enter your age  
20  
you are eligible to vote  
vboxuser@hello:~$ █
```



Q2. Script to reverse a string.

Reversing a string

s = "Hello"

strlen = \${#s}

for ((i = \${strlen}-1; i >= 0; i--));

do

reverse = \$reverse\${s:\$i:1}

done

echo "Original String: \$s"

echo "Reversed String: \$reverse"

Q3. Script to read age of a person & display if he is eligible to vote

#!/bin/bash

echo "enter your age"

read age

if ["\$age" -ge 18];

then

 echo "you are eligible to vote"

else

 echo "you are younger!!"

fi

```
vboxuser@hello:~/Color$ ./Colors.sh  
enter color  
red  
I like red color  
vboxuser@hello:~/Color$ ./Colors.sh  
enter color  
black  
Sorry, colors available are red, blue, green, yellow  
vboxuser@hello:~/Color$
```

Q. Script to demonstrate switch statement to
give color of your choice.

echo "Enter color"

read color

case \$color in
blue)

 echo "I like blue color"

;;

 green)

 echo "I like green color"

;;

 red)

 echo "I like red color"

;;

 yellow)

 echo "I like yellow color"

;;

*)

 echo "Sorry, color available are red, blue,
 green, yellow"

;;

esac

```
Sorry, colors available are red, blue, green, yellow  
vboxuser@hello: $ ./Democat.sh  
Enter file name: Age.sh  
#!/bin/bash  
echo "enter your age"  
read age  
if [ "$age" -ge 18 ];  
then  
echo "you are eligible to vote"  
else  
echo "you are younger!!"  
fi
```

: \$

vboxuser@hello: \$

```
vboxuser@hello: ~ $ ./Numbers.sh
```

1
2
3
4
5
6
7
8
9
10

vboxuser@hello: ~ \$

Q5. Read the contents of a file & display the same on screen

read -p "Enter file name:" filename

while read line

do

echo \$line

done < \$filename

Q6. Display 10 numbers using for loop.

for i in 1 2 3 4 5 6 7 8 9 10

do

echo \$i

done

2/10

WEEK-08

AUTOMATION OF SYSTEM TASKS

Illustrate automation of basic tasks like monitoring memory consumption, check connectivity etc at different frequencies.

The cron is a shell utility, offered by a Linux like OS that automates the scheduled task at a predetermined time.

Creating cron jobs

- * To create or edit a cron job as the root user, run the command. # crontab -e

Listing crontab files

- * To view the cron jobs that have been created, simply pass the -l option as shown # crontab -l

Deleting a crontab file

- * To delete a cron file, simply run crontab -e and delete or the line of the cronjob that user wants and save the file. # crontab -e.

The rules which govern the format of date & time field as follows:-

- * When any of the first five fields are set to an asterisk (*), it stands for all the values of the field. For instance, to execute a command daily, we can put an asterisk (*) in the week's field.
- * One can also use a range of numbers, separated with a hyphen (-) in the time & date field to include more than one contiguous value but not all the values of the field. For example, we can use the 7-10 to run a cmd from July to October.

Permitting users to run cron job:-

- * The user must be listed in this file to be able to run cron jobs if the file exists.
/etc/cron.allow
- * If the ~~cron.allow~~ file doesn't exist but the ~~cron.deny~~ file exists, then a user must not be listed in this file to be able to run the cron job.
/etc/cron.deny

CRON TABLE FORMAT

*	*	*	*	*	Command-to-execute
1	1	1	1	1	Day of the Week (0-6) (Sunday=0)
1	1	1	1	1	Month (1-12)
1	1	1			Day of Month (1-31)
1		1			Hour (0-23)
1					Min (0-59)

To install sudo apt-get install cron

Setting up the Cron Operation

We need to open the crontab along with a text editor for configuring the cron operation and enter the syntax for a command we wish to execute.

Example 1:- crontab -e

Choose 1 in the option.

Syntax:

* * * * * echo "Hello" >> /tmp/hello.txt

Save and exit.

\$ cat /tmp/hello.txt

Example 2:- To run a job at 4.35 pm edit crontab

Syntax: crontab -e

35 16 * * * echo "Hello from cronjob" >> /tmp/hello1.txt

Save and ~~exit~~ exit

\$ cat /tmp/hello1.txt

```
not reachable  
root@xenial:~$ cd /tmp/  
root@xenial:~/tmp$ gedit memcon.sh  
root@xenial:~/tmp$ chmod 777 memcon.sh  
root@xenial:~/tmp$ crontab -e  
crontab: installing new crontab  
root@xenial:~/tmp$ cat /tmp/memcon.txt  
cat: /tmp/memcon.txt: No such file or directory  
root@xenial:~/tmp$ cat /tmp/memcon.txt  
cat: /tmp/memcon.txt: No such file or directory  
root@xenial:~/tmp$ cd tmp  
bash: cd: tmp: No such file or directory  
root@xenial:~/tmp$ cd /tmp  
root@xenial:/tmp$ cat memcon.txt  
memory consumption is as below  
root@xenial:/tmp$ cat memory.txt  
Mem:       total        used        free      shared  buff/cache   available  
Swap:      2006280     758756    279012      37968      976512    1869188  
root@xenial:/tmp$
```

Check remote server connectivity using cronjob
Save below file as ping.sh

Syntax: gedit ping.sh

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

```
ping -c "google.com" >/dev/null
```

```
if [ $? -eq 0 ] then
```

```
echo "reachable"
```

```
exit 0
```

else

```
echo "not reachable"
```

fi

Save and exit.

```
$ crontab -e
```

```
* * * * * /home/ubuntu/ping.sh >>/tmp/ping.txt
```

Save and exit

```
$ cat /tmp/ping.txt
```

~~Example 3: Automation script to check memory consumption~~

Syntax: gedit memon.sh

echo "memory consumption is as below"

```
free > /tmp/memory.txt
```

Save and exit.

```
$ crontab -e
```

```
* * * * * /home/ubuntu/memon.sh
```

```
$ cat /tmp/memory.txt
```

```
root@hello:/home/nandu/Desktop# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.189.214 netmask 255.255.255.0 broadcast 192.168.189.255
      inet6 fe80::537f:987c:b1c2:e978 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:a4:ed:16 txqueuelen 1000 (Ethernet)
          RX packets 22521 bytes 31909293 (31.9 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 5323 bytes 385913 (385.9 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
          RX packets 154 bytes 14156 (14.1 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 154 bytes 14156 (14.1 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
root@hello:/home/nandu/Desktop# iwconfig
lo      no wireless extensions.

enp0s3  no wireless extensions.
```

WEEK-09 NETWORK MANAGEMENT.

Enable internet on Linux VM.

01. Ensure the internet connection.
02. Turnoff the virtual machine.
03. Open network settings of VM.
04. Enable the Network Adapter.
05. Select the Network Device.
06. Save settings & start the VM.

* ifconfig

- ifconfig (interface configuration) is used to configure the kernel-resident network interface.
- It is used at the boot time to set up the interfaces as necessary.

* iwconfig

- iwconfig cmd in Linux is like ifconfig cmd, in the sense it works with kernel-resident network interface but it is dedicated to wireless networking interfaces only.

```
Proceeding triggers for main-00 (2.10.2-2) ...
root@hell0:/home/nandu/Desktop# ethtool enp0s3
Settings for enp0s3:
Supported ports: [ TP ]
Supported link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Supported FEC modes: Not reported
Advertised link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Advertised FEC modes: Not reported
Speed: 1000Mb/s
Duplex: Full
Auto-negotiation: on
Port: Twisted Pair
PHYAD: 0
Transceiver: Internal
MDI-X: off (auto)
Supports Wake-on: unbg
Wake-on: d
Current message level: 0x00000007 (7)
                         drv probe link

Link detected: yes
10:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
```

```
root@hello:/home/nandu/Desktop# arp  
Address          Hwtype  Hwaddress      Flags Mask Iface  
_gateway        ether    92:6c:14:af:73:68  C  
root@hello:/home/nandu/Desktop#
```

* ethtool

The ethtool utility on Linux allows you to view and change some of your network-driver & interface-card settings, especially for wired devices.

sudo apt install ethtool

* arpwatch

ARP also known as "Address Resolution Protocol" is a very useful tool to find IP to MAC address mappings.

* telnet

The telnet cmd is used to create a remote connection with a system over a TCP/IP network. It allows us to administrate other systems by the terminal.

* netstat

~~netstat is a command line utility for Linux that prints network connections, routing tables, interface statistics and multipath membership~~

```
[root@hell0 ~]# ping -c 10 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=111 time=69.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=111 time=76.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=111 time=63.4 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=111 time=78.7 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=111 time=78.4 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=111 time=76.1 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=111 time=75.7 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=111 time=73.2 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=111 time=70.9 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=111 time=68.9 ms

--- 8.8.8.8 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9019ms
rtt min/avg/max/mdev = 63.372/73.121/78.692/4.666 ms
[root@hell0 ~]
```

```
user@user-OptiPlex-5090:~$ traceroute www.google.com
traceroute to www.google.com (142.250.196.30), 30 hops max, 60 byte packets
  1 gateway (192.168.189.32)  3.999 ms  3.994 ms  2.574 ms
  2
  3  192.168.19.39 (192.168.19.39)  61.321 ms  56.8.48.49 (56.8.48.49)  68.884 ms  56.8.48.13 (56.8.48.13)  61.676 ms
  4  192.168.0.55.248 (192.168.0.55.248)  69.871 ms  192.168.0.55.248 (192.168.0.55.248)  39.865 ms  192.168.0.55.252 (192.168.0.55.252)  68.219 ms
  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
```

* ping

ping (packet internet groper) cmd is used to check the network connectivity between host and server/host.

* traceroute

traceroute cmd in linux system prints the route that a packet takes to reach the host.

* tcpdump

tcpdump is a packet sniffing & packet analyzing tool for a System Administrator to troubleshoot connectivity issues in linux.

It is used to capture, filter & analyze network traffic such as TCP/IP packets going through system.

S (10)

```
root@hello:/home/nandu/Desktop# useradd nandini  
root@hello:/home/nandu/Desktop# passwd nandini  
New password:  
Retype new password:  
passwd: password updated successfully  
root@hello:/home/nandu/Desktop#
```

?

```
root@hello:/home/nandu/Desktop# useradd nandini  
useradd: user 'nandini' already exists  
root@hello:/home/nandu/Desktop# deluser nandini  
Removing user 'nandini' ...  
Warning: group 'nandini' has no more members.  
Done.
```

WEEK - 10

USER AUTHENTICATION

* useradd

- useradd is a cmd in Linux that is used to add user accounts to system.
- When we run the "useradd" command in the Linux terminal, it performs the foll major things:-
useradd username.

* passwd

- The passwd cmd changes passwords for user accounts.
- A normal user may only change the password for their own account, while the superuser may change the password for any account.

* userdel

- userdel cmd in Linux system is used to delete a user account and related files.
- This cmd basically modifies the system account files, deleting all the entries which refer to the username LOGIN
userdel username.

```
root@hello:/home/nandu/Desktop# usermod -d /home/nn nandini  
root@hello:/home/nandu/Desktop# cat /etc/passwd|grep nandini  
nandini:x:1003:1003::/home/nn:/bin/sh  
root@hello:/home/nandu/Desktop# groupadd cssjp  
root@hello:/home/nandu/Desktop# tail /etc/group  
geoclue:x:131:  
pulse:x:132:  
pulse-access:x:133:  
gdm:x:134:  
lxd:x:135:nandu  
vboxuser:x:1000:vboxuser  
sanbashare:x:136:nandu  
nandu:x:1002:  
nandini:x:1003:  
cssjp:x:1004:
```

```
root@hello:/home/nandu/Desktop# addgroup sjp  
Adding group 'sjp' (GID 1001) ...  
Done.  
root@hello:/home/nandu/Desktop# tail /etc/group  
pulse:x:132:  
pulse-access:x:133:  
gdm:x:134:  
lxd:x:135:nandu  
vboxuser:x:1000:vboxuser  
sanbashare:x:136:nandu  
nandu:x:1002:  
nandini:x:1003:  
cssjp:x:1004:  
sjp:x:1001:  
root@hello:/home/nandu/Desktop# usermod -g sjp nandini  
root@hello:/home/nandu/Desktop# groupmod -n grpnew cssjp  
root@hello:/home/nandu/Desktop# tail /etc/group  
pulse:x:132:  
pulse-access:x:133:  
gdm:x:134:  
lxd:x:135:nandu  
vboxuser:x:1000:vboxuser  
sanbashare:x:136:nandu  
nandu:x:1002:  
nandini:x:1003:  
sjp:x:1001:  
grpnew:x:1004:
```

```
root@hello:/home/nandu/Desktop# groupadd cssjp  
root@hello:/home/nandu/Desktop# tail /etc/group  
geoclue:x:131:  
pulse:x:132:  
pulse-access:x:133:  
gdm:x:134:  
lxd:x:135:nandu  
vboxuser:x:1000:vboxuser  
sanbashare:x:136:nandu  
nandu:x:1002:  
nandini:x:1003:  
cssjp:x:1004:  
root@hello:/home/nandu/Desktop# usermod -g cssjp nandini
```

* usermod

- usermod cmd for modify user is used to change the properties of a user.
- To change the home directory of a user
 usermod -d /home/newfolder existinguser.
 usermod -g #hongroupname existinguser.

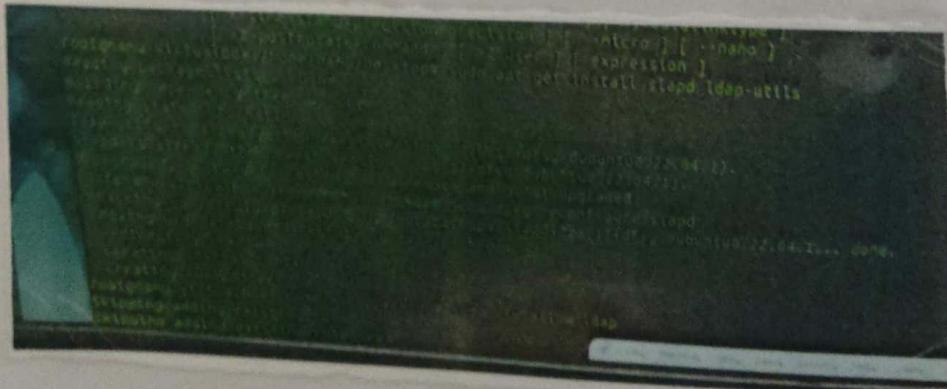
* groupadd

- groupadd cmd is used to create a new user group.
- groupadd newgroupname.
- Every new group registered in the file "/etc/group".
 sudo tail /etc/group.
 usermod -g existinggroupname existinguser.

* groupmod

- groupmod cmd is used to modify or change the existing group.
 - It can be handled by superuser or root user.
- groupmod groupname
 groupmod -n groupnewname groupoldname.
 tail /etc/group

```
grpnew:x:1004:  
root@hello:/home/nandu/Desktop# gpasswd grpnew  
Changing the password for group grpnew  
New Password:  
Re-enter new password:  
root@hello:/home/nandu/Desktop# groupdel sjp  
groupdel: cannot remove the primary group of user 'nandint'  
root@hello:/home/nandu/Desktop# addgroup vmln  
Adding group 'vmln' (GID 1005) ...  
Done.  
root@hello:/home/nandu/Desktop# delgroup vmln  
Removing group 'vmln' ...  
Done.  
root@hello:/home/nandu/Desktop# tail /etc/group  
pulse:x:132:  
pulse-access:x:133:  
gdm:x:134:  
lxd:x:135:nandu  
vboxuser:x:1000:vboxuser  
sanbashare:x:136:nandu  
nandu:x:1002:  
nandint:x:1003:  
sjp:x:1001:  
grpnew:x:1004:  
root@hello:/home/nandu/Desktop# top
```



* igpasswd

- `igpasswd` cmd is used to administer the `letc/group`
- `igpasswd` cmd assigns a user to a group with some security criteria.

igpasswd group

* groupdel

- `groupdel` cmd is used to delete a existing group.
- It will delete all entry that refers to the group, modifies the system account files, and it is handled by superuser or root user.

groupdel existinggroup

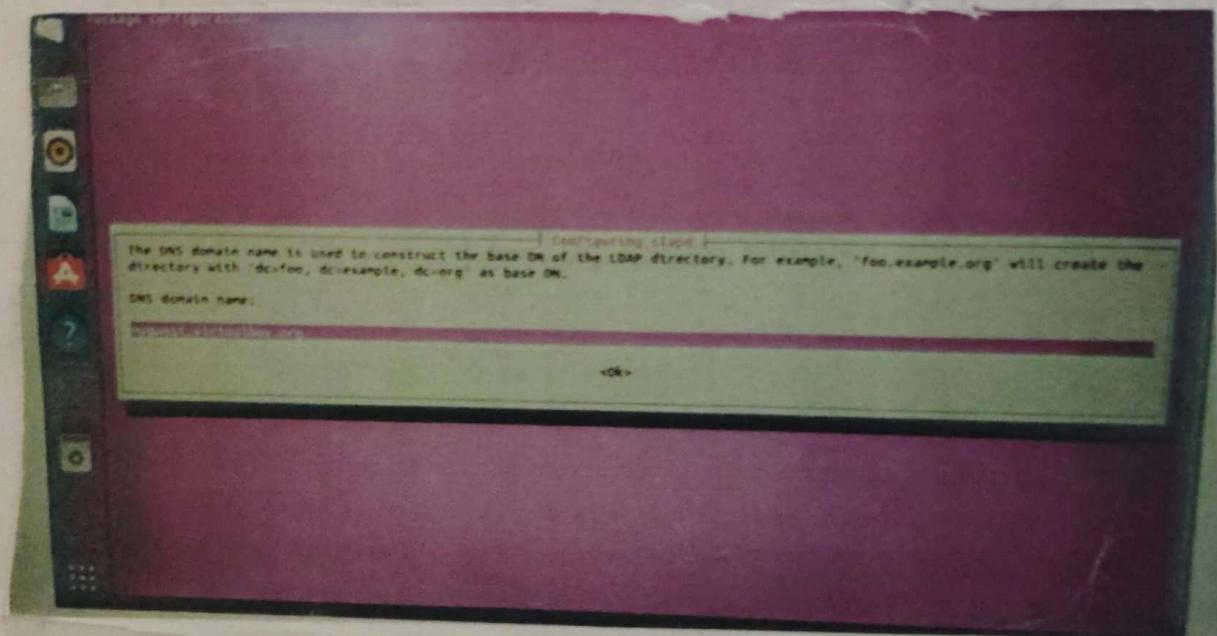
LDAP server and client configuration-

- Lightweight Directory Access Protocol is a standard protocol designed to manage & access hierarchical directory information over a network.

Installing and configuring the LDAP Server.

- Install the LDAP server & some associated utilities.
- `sudo apt-get update`
- `sudo apt-get install slapd ldap-utils`.
- `sudo dpkg-reconfigure slapd`.

```
Creating LDAP directory... done.
root@namu-VirtualBox:/home/namu/Desktop$ sudo ufw allow ldap
Skipping adding existing rule
Skipping adding existing rule
root@namu-VirtualBox:/home/namu/Desktop$ 
root@namu-VirtualBox:/home/namu/Desktop$ 
root@namu-VirtualBox:/home/namu/Desktop$ sudo slapadd -F /etc/openldap/ldif -H ldap:// -x
anonVNOUS
root@namu-VirtualBox:/home/namu/Desktop$ 
```



There were quite a few new question to answer this process.

- For the first option select NO.
- Then type the domain name & set password.
- Then organization name and set password.
- Then select MP B.
- Then select No, Yes, No option for the questions.
- sudo su -u allow ldap
- Test LDAP connection with ldapwhoami
ldapwhoami -H ldap://1 -x

✓ (10)

```
root@nvidia:~# top
```

Top - 22:49:37 up 39 min, 3 users, load average: 0.30, 0.23, 0.04
Tasks: 180 total, 3 running, 180 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0 us, 0.0 sy, 0.0 id, 99.8 sl, 0.0 wa, 0.0 hi, 0.0 st, 0.0 et
Mem: 1998.4 total, 133.3 free, 766.2 used, 1122.9 buff/cache
Swap: 0 total, 2870.7 free, 0.0 used, 3847.8 avail swap

Tasks	Time	User	System	Cpu(s)	Memory	Swap	Process
1744	0:00	0	0	0.0	11.3	1.0	gnome-shell
56	0:00	0	0	0.0	0.0	0.0	konsole/0.2 events
305	0:00	0	0	0.0	0.0	0.0	systemd-coredump
1784	0:00	0	0	0.0	2.7	0.0	gnome-terminal
4386	0:00	0	0	0.0	0.0	0.0	konsole/0.2 events_unbound
4029	0:00	0	0	0.0	0.0	0.0	top
1	0:00	0	0	0.0	0.0	0.0	systemd
2	0:00	0	0	0.0	0.0	0.0	kernel-thread
3	0:00	0	0	0.0	0.0	0.0	rcu_gop
4	0:00	0	0	0.0	0.0	0.0	rcu_par_gop
5	0:00	0	0	0.0	0.0	0.0	slab.Flushed
6	0:00	0	0	0.0	0.0	0.0	netns
7	0:00	0	0	0.0	0.0	0.0	net_pcpus_wq
8	0:00	0	0	0.0	0.0	0.0	rcu_tasks_stopped
9	0:00	0	0	0.0	0.0	0.0	rcu_tasks_node_kthread
10	0:00	0	0	0.0	0.0	0.0	rcu_tasks_node_stopped
11	0:00	0	0	0.0	0.0	0.0	kernel-thread
12	0:00	0	0	0.0	0.0	0.0	rcu_dereference
13	0:00	0	0	0.0	0.0	0.0	rcu_dereference_stopped
14	0:00	0	0	0.0	0.0	0.0	rcu_bh
15	0:00	0	0	0.0	0.0	0.0	rcu_bh_bh
16	0:00	0	0	0.0	0.0	0.0	rcu_bh_bh_bh
17	0:00	0	0	0.0	0.0	0.0	rcu_bh_bh_bh_bh
18	0:00	0	0	0.0	0.0	0.0	rcu_bh_bh_bh_bh_bh
19	0:00	0	0	0.0	0.0	0.0	cpuhot
20	0:00	0	0	0.0	0.0	0.0	ideventsfs
21	0:00	0	0	0.0	0.0	0.0	mem_cgroup_MG
22	0:00	0	0	0.0	0.0	0.0	auditd
23	0:00	0	0	0.0	0.0	0.0	hungtrapd
24	0:00	0	0	0.0	0.0	0.0	dm_crypt
25	0:00	0	0	0.0	0.0	0.0	writeback
26	0:00	0	0	0.0	0.0	0.0	dcopcontrol
27	0:00	0	0	0.0	0.0	0.0	lvm
28	0:00	0	0	0.0	0.0	0.0	md
29	0:00	0	0	0.0	0.0	0.0	md

```
root@hello:/home/nando/Desktop# lstat  
Linux 6.2.0-26-generic (hello) 25/04/24 x86_64 (1 CPU)
```

avg-cpu: User	Nice	System	Swait	Sidle			
5.77	0.51	3.41	0.44	0.00	93.87		
Device	tps	kB_read/s	kB_wrtn/s	kB_dcd/s	kB_read	kB_wrtn	kB_dcd
loop0	0.00	0.00	0.00	0.00	0	0	0
loop1	0.01	0.15	0.00	0.00	0	0	0
loop18	0.02	0.11	0.00	0.00	364	0	0
loop21	0.01	0.09	0.00	0.00	371	0	0
loop12	0.10	6.39	0.00	0.00	334	0	0
loop13	0.00	0.00	0.00	0.00	2229	0	0
loop2	0.10	1.66	0.00	0.00	14	0	0
loop3	0.01	0.10	0.00	0.00	4020	0	0
loop4	0.02	0.34	0.00	0.00	349	0	0
loop5	0.02	0.32	0.00	0.00	2025	0	0
loop6	0.02	0.33	0.00	0.00	1874	0	0
loop7	0.05	4.78	0.00	0.00	2385	0	0
loop8	0.19	2.80	0.00	0.00	15757	0	0
loop9	0.01	0.10	0.00	0.00	4093	0	0
loop10	0.11	432.00	262.48	0.00	340	0	0
					774063	431000	

WEEK-11

* top

top cmd is a performance monitoring program that can display CPU usage, memory usage, swap memory, cache size, buffer size, process PID, users, emails etc.

* iostat

The iostat cmd in linux lets you monitor CPU utilization & I/O statistics of all the disk & file systems.

iostat cmds are useful for changing system configuration to better balance the io load between physical disks.

* cat /proc/cpuinfo

To see what type of processor (CPU) your computer system has, use the Linux cmd: \$ cat /proc/cpuinfo.

```
root@hello:/home/nandu/Desktop# cat /proc/meminfo
MemTotal:       2086628 kB
MemFree:        161580 kB
MemAvailable:   1905278 kB
Buffers:         32400 kB
Cached:          634400 kB
SwpCached:      562094 kB
ActiveVM:       980430 kB
InactiveVM:     3488 kB
Active(anon):   607249 kB
Inactive(anon): 538595 kB
Active(File):   379196 kB
Inactive(File): 0 kB
Unevictable:    0 kB
Klocked:        0 kB
SwapTotal:      2744316 kB
SwapFree:       2737612 kB
Zswap:          0 kB
Zswapped:       0 kB
Dirty:           124 kB
Writeback:      0 kB
AnonPages:      572524 kB
Mapped:          281076 kB
Shmem:          37024 kB
Kmceklmable:    60476 kB
Slab:            172232 kB
Kmceklmable:    60476 kB
Sunreclaim:     91756 kB
KernelStack:    6796 kB
PageTables:     14928 kB
SecPageTables:  0 kB
NFS_Unstable:   0 kB
Bounce:          0 kB
MMLTAbckTmp:   0 kB
CommitLimit:    3747628 kB
Committed_AS:  3595500 kB
AvailableMemory: 14559738367 kB
```

```
root@hello:/home/nandu/Desktop# free
total        used        free      shared  buff/cache available
Mem:    2086628      789652      161580      37824    1055396  1088380
Swap:   2744316       6784    2737612
root@hello:/home/nandu/Desktop# df
Filesystem  1K-blocks  Used Available Use% Mounted on
tmpfs        208664    1464    199200  1% /run
/dev/sda3    25106692 11884716  11921292  50% /
tmpfs        1003312      0    1003312  0% /dev/shm
tmpfs        5120        4      5116  1% /run/lock
/dev/sda2     524252    6216    518036  2% /boot/efi
tmpfs        208660     116    208544  1% /run/user/1002
root@hello:/home/nandu/Desktop# df /dev/sda3
Filesystem  1K-blocks  Used Available Use% Mounted on
/dev/sda3    25106692 11884716  11921292  50% /
root@hello:/home/nandu/Desktop# df aa.txt
df: aa.txt: No such file or directory
```

* cat /proc/meminfo

The '/proc/meminfo' is used to report the amount of free & used memory on the system as well as the shared memory & buffers used by the kernel.

* free

free cmd which displays the total amount of free space available along with the amount of memory used & swap memory in the system, & also the buffers used by the kernel.

* df

df cmd that displays the amount of disk space available on the file system containing each file name argument.

```
[root@hello:/home/nandu/Desktop]# dmesg
Linux version 6.2.0-26-generic (buildd@nantes03-nd64-042) (x86_64-linux-gnu-gcc-11 (Ubuntu 11.3.0-1ubuntu3-22.04.1)-11)
  ., GNU ld (GNU Binutils for Ubuntu) 2.38) #26~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Jul 13 10:27:29 UTC 2 (Ubuntu 6.2.0-26-22.04.1)
Command Line: BOOT_IMAGE=/boot/vmlinuz-6.2.0-26-generic root=UUID=d8cc9fe2-970d-4980-ab4c-bb5a07a2887d ro quiet splash
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel Conroe/Intel
[    0.000000]   AMD AUTHENTICAMD
[    0.000000]   Hygon HygonGenzine
[    0.000000]   Centaur CentaurHavila
[    0.000000]   Phoenix Shanghai
[    0.000000] x86/pov MMU will use FXSAVE
[    0.000000] signals: Max stackframe size: 1440
[    0.000000] BIOS-provided physical RAM map:
[    0.000000]   BIOS-e820: [mem 0000000000000000-000000000000ffff] usable
[    0.000000]   BIOS-e820: [mem 000000000009fc00-00000000000ffff] reserved
[    0.000000]   BIOS-e820: [mem 0000000000000000-00000000000ffff] reserved
[    0.000000]   BIOS-e820: [mem 0000000000000000-00000000000ffff] usable
[    0.000000]   BIOS-e820: [mem 0000000000000000-00000000000ffff] NCPI data
[    0.000000]   BIOS-e820: [mem 0000000000000000-00000000000ffff] reserved
[    0.000000] NX (Execute disable) protection: active
[    0.000000] SMBIOS 2.5 present.
[    0.000000] DRAM: Innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[    0.000000] Hypervisor detected: KVM
[    0.000000] kvm-clock: Using mcrs 4b504d01 and 4b504d00
[    0.000000] kvm-clock: Using sched offset of 6598876805 cycles
[    0.000000] clocksource: kvm-clock: mask: 0xfffffffffffffff max_cycles: 0x1cd42e4dff, max_idle_ns: 801500591403 ns
[    0.000000] tsc: Detected 2499.998 MHz processor
[    0.000000] e820: update [mem 0x000000000-0x0000000ffff] usable => reserved
[    0.000000] e820: remove [mem 0x000000000-0x0000000ffff] usable
[    0.000000] last_pfn = 0x7ffff Max_arch_pfn = 0x400000000
```

```
[root@hello:/home/nandu/Desktop]# cd /var/log
[root@hello:/var/Log]# ls
alternatives.log  bootstrap.log  dmesg.0      dpkg.log      hp          openvpn      ubuntu-advantage.log
apport.log        btmp          dmesg.1.gz   faillog      installer   private      unattached-upgrade.log
auth.log          cups          dmesg.2.gz   fontconfig.log journal     speech-dispatcher
boot.log          dmesg         dmesg.3.gz   gdm          kern.log    syslog      vboxpostinstall.log
root@hello:/var/Log# sudo head boot.log
.... Mon Feb 21 12:34:49 IST 2024 ....
[  OK ] Started Show Plymouth Boot Screen.
[  OK ] Started Forward Password R.A to Plymouth Directory Watch.
[  OK ] Reached target Local Encrypted Volumes.
[  OK ] Listening on Load/Save MP .itch Status /dev/rfkill Watch.
[  OK ] Found device VBOX_HARDDISK_EFI|x20System|x20Partition.
Starting File System Check on /dev/disk/by-uuid/0E90-4461...
[  OK ] Started File System Check daemon to report status.
[  OK ] Finished File System Check on /dev/disk/by-uuid/0E90-4461.
Mounting /boot/efi...
root@hello:/var/Log# cd
root@hello:-# cd /home/nandu/Desktop
root@hello:/home/nandu/Desktop #
```

* dmesg

This cmd, shorthand for "diagnostic message", is a Linux utility for displaying the messages that flow within the kernel using a buffer.

* Shutdown

The shutdown cmd in Linux is used to shutdown the system in a safe way. You can shutdown the machine immediately.

\$ shutdown now.

* cd /var/log

You can view the contents of this directory by issuing the following cmd.

\$ ls /var/log

(10)

* halt

This cmd in Linux is used to instruct the hardware to stop all the CPU functions.

\$ halt.

```
root@hello:/home/nandu/Desktop# sudo apt install ftp
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ftp is already the newest version (20210827-4build1).
ftp set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 279 not upgraded.
root@hello:/home/nandu/Desktop# sudo apt install vsftpd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  vsftpd
  0 upgraded, 1 newly installed, 0 to remove and 279 not upgraded.
Need to get 123 kB of archives.
After this operation, 326 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 vsftpd amd64 3.0.5-ubuntu1 [123 kB]
Fetched 123 kB in 3s (44.2 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 101903 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.5-ubuntu1_amd64.deb ...
Unpacking vsftpd (3.0.5-ubuntu1) ...
Setting up vsftpd (3.0.5-ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /lib/systemd/system/vsftpd.service
Processing triggers for man-db (2.10.2-1) ...
```

```
root@hello:/home/nandu/Desktop# chmod 777 /etc/vsftpd.conf
root@hello:/home/nandu/Desktop# nano /etc/vsftpd.conf
root@hello:/home/nandu/Desktop# systemctl restart vsftpd
```

```
17:51:00 prompt:      127.0.0.1 21
root@hello:/home/nandu/Desktop# ftp 127.0.0.1 21
Connected to 127.0.0.1.
220 (vsFTPd 3.0.5)
Name (127.0.0.1:nandu): nandu
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get /home/nandu/Desktop/a.txt test.txt
local: test.txt remote: /home/nandu/Desktop/a.txt
229 Entering Extended Passive Mode (|||52712|)
150 Opening BINARY mode data connection for /home/nandu/Desktop/a.txt (0 bytes).
  0     0.00 KiB/s
226 Transfer complete.
ftp> put /home/nandu/Desktop/test.txt b.txt
local: /home/nandu/Desktop/test.txt remote: b.txt
229 Entering Extended Passive Mode (|||52610|)
150 Ok to send data.
  0     0.00 KiB/s
226 Transfer complete.
ftp>
```

WEEK-12

FTP server on LINUX & transfer files to demonstrate its working.

ftp is the user interface to standard File Transfer Protocol. The program allows a user to transfer files to & from a remote network site.

Prerequisites:

- * Install Ftp & vsftpd using software Manager.
- * Change /etc/vsftpd.conf file to enable write permission (chmod 777 /etc/vsftpd.conf).
- * nano /etc/vsftpd.conf
- * Search for #write_enable=YES (around 30th line)
- * Uncomment & Save & exit
- * Restart the service systemctl restart vsftpd

Example 1: Get the connection to ftp

- * ftp 127.0.0.1 21 /22.
- * Enter user name & password for connection, this will take user to ftp utility & then issue ftp set of cmd.

```
Aug 10 22:04  
root@nemu:~# apt-get update  
[sudo] password for nemu:  
Get:1 http://ja.yankeesecurity.net/ubuntu security/Release [219 kB]  
Get:2 http://ja.yankeesecurity.net/ubuntu security/main Sources [304 kB]  
Get:3 http://ja.yankeesecurity.net/ubuntu security/main Packages [10,000 kB]  
Get:4 http://ja.yankeesecurity.net/ubuntu security/restricted Packages [100 kB]  
Get:5 http://ja.yankeesecurity.net/ubuntu security/universe Packages [100 kB]  
Get:6 http://ja.yankeesecurity.net/ubuntu security/universe Sources [100 kB]  
Get:7 http://ja.yankeesecurity.net/ubuntu security/universe Translation-en [100 kB]  
Get:8 http://ja.yankeesecurity.net/ubuntu main Sources [1000 kB]  
Get:9 http://ja.yankeesecurity.net/ubuntu main Packages [1000 kB]  
Get:10 http://ja.yankeesecurity.net/ubuntu universe Sources [1000 kB]  
Get:11 http://ja.yankeesecurity.net/ubuntu universe Packages [1000 kB]  
Get:12 http://ja.yankeesecurity.net/ubuntu universe Translation-en [1000 kB]  
Get:13 http://ja.yankeesecurity.net/ubuntu multiverse Sources [1000 kB]  
Get:14 http://ja.yankeesecurity.net/ubuntu multiverse Packages [1000 kB]  
Reading Package Lists... done  
root@nemu:~# /etc/init.d/NetworkManager start apache2  
root@nemu:~# /etc/init.d/NetworkManager enable apache2  
root@nemu:~# /lib/systemd/system-set-default enable apache2  
root@nemu:~# /etc/init.d/NetworkManager restart apache2  
root@nemu:~# cd /var/www/html  
root@nemu:~# nano index.html  
root@nemu:~# ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.109.214 netmask 255.255.255.0 broadcast 192.168.109.255  
inet6 fe80::1d50:9699%enp0s3: prefixlen 64 scopeid 0x20<link>  
ether 08:00:27:da:25:f3 txqueuelen 1000 (Ethernet)  
RX packets 6466 bytes 8366965 (8.3 MB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 2384 bytes 264469 (264.4 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
root@nemu-VirtualBox:~# Desktop5 sudo su  
[sudo] password for nemu:  
root@nemu-VirtualBox:~# mkdir /var/www/aas  
root@nemu-VirtualBox:~# cd /var/www/aas  
root@nemu-VirtualBox:~/var/www/aas# nano index.html  
root@nemu-VirtualBox:~/var/www/aas# ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
inet6 fe80::1d50:9699%enp0s3: prefixlen 64 scopeid 0x20<link>  
ether 08:00:27:da:25:f3 txqueuelen 1000 (Ethernet)  
RX packets 106 bytes 55098 (55.0 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 150 bytes 31994 (31.9 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Local Loopback)  
RX packets 118 bytes 10072 (10.0 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 118 bytes 10072 (10.0 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
root@nemu-VirtualBox:~/var/www/aas# cd  
root@nemu-VirtualBox:~# cd /home/nemu/Desktop
```

Example 2: Download the file from ftp server

- get /home/nandu/Desktop/a.txt test.txt
will download the file a.txt from ftp server to current user Desktop with file name test.txt.

Example 3: Upload the file from user@server

- put /home/nandu/Desktop/test.txt index.html test.txt

Apache Server Setup

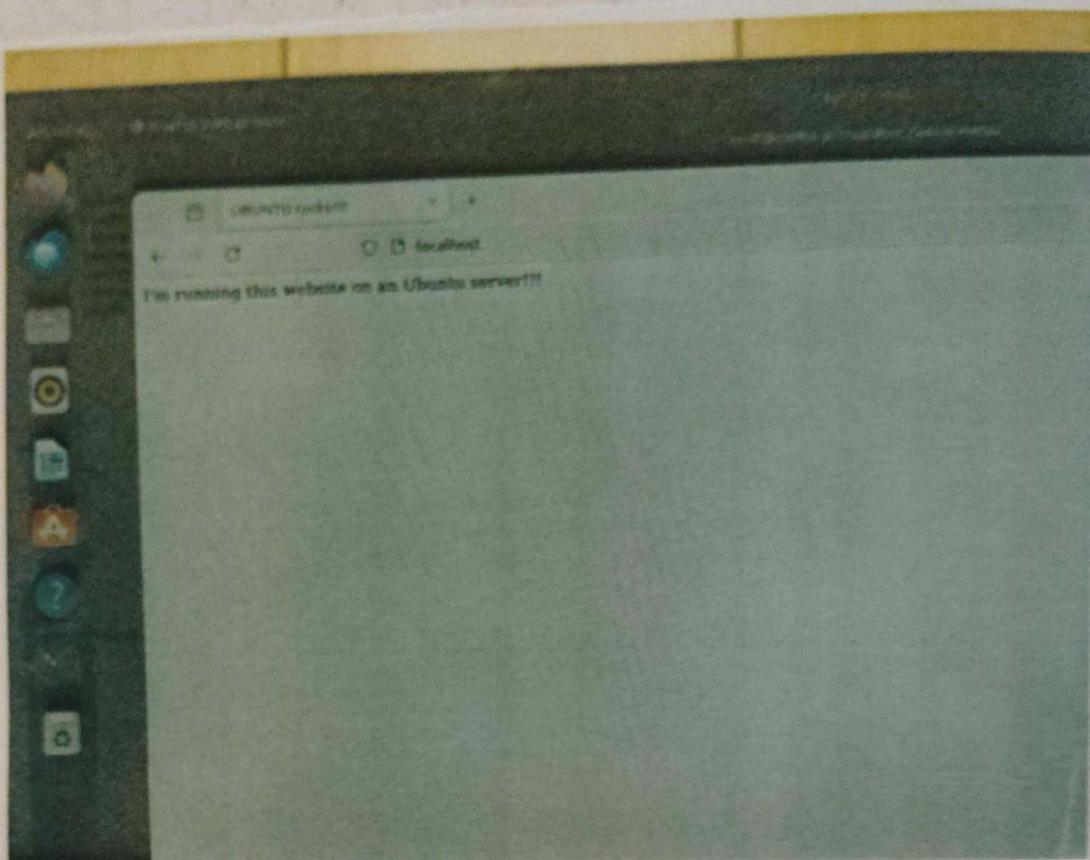
To install Apache, install the latest meta-package apache2 by running:

```
sudo apt update  
sudo apt install apache2
```

After letting the command run, all required packages are installed & we can test it out by typing in our IP address for the web server.

To change default page from normal page to custom page create html file in /etc/var/html/test.html

```
<html>  
<head>  
<title> UBUNTU rocks !!! </title>  
</head>  
<p> I'm running this website on an Ubuntu Server!!! </p>  
<body>  
</html>
```



The screenshot shows a Firefox browser window with the URL "127.0.0.1" in the address bar. The main content is the "Apache2 Default Page". It features the Ubuntu logo and the text "Ubuntu". To the right, there is a red button with the white text "It works!". Below the logo, there is a paragraph of text explaining the purpose of the page. At the bottom, there is a section titled "Configuration Overview" with some technical details about the Apache2 configuration files.

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in [Ansible](#), [/usr/share/doc/apache2/README.Debian.gz](#). Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.

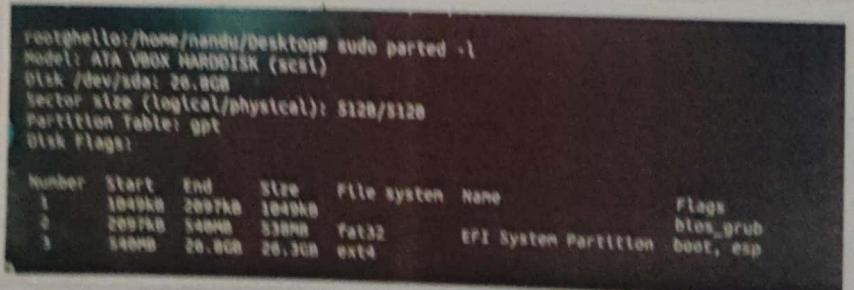
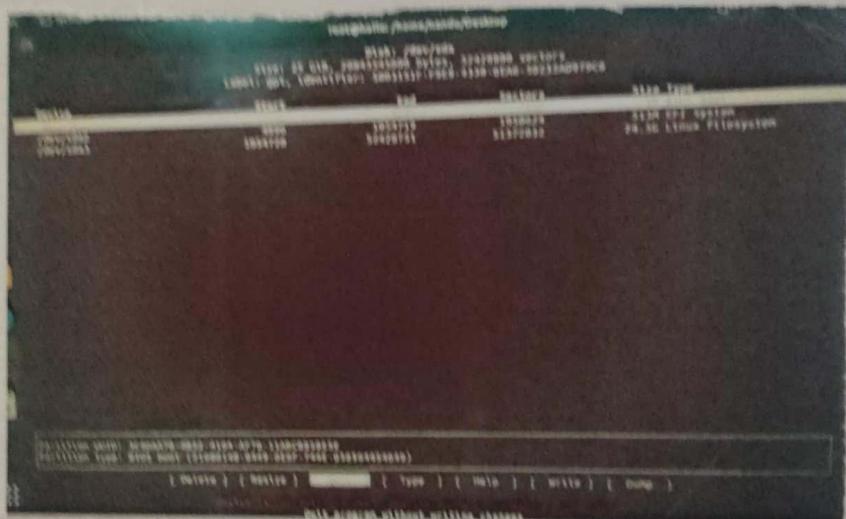
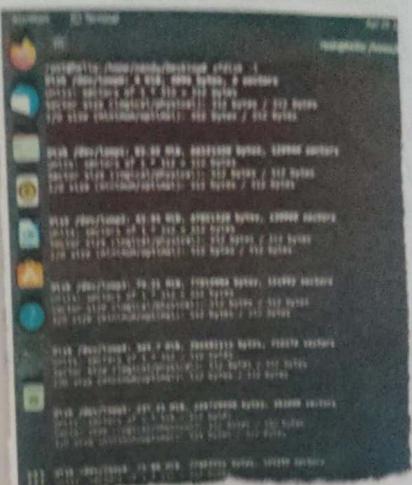
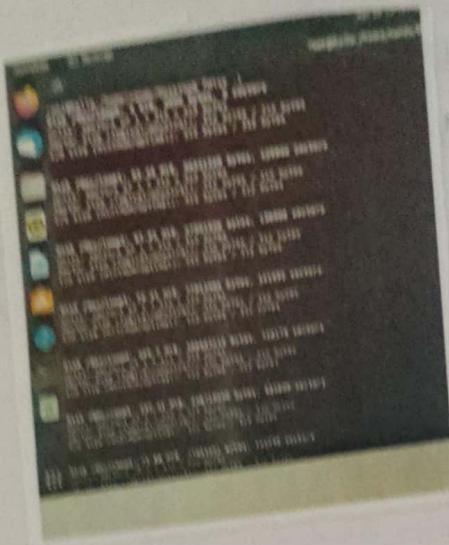
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- parts.conf
|   |-- mods-enabled
|       |-- *.load
|       |-- *.conf
|   |-- conf-enabled
|       |-- *.conf
|   |-- sites-enabled
|       |-- *.conf
```

Now go to /etc/apache2/mods-enabled directory
& edit configuration file
nano dir.conf

Change index.html page to your page.
Open browser & type localhost, it displays your
created html page.

✓ 10



WEEK - 13

STORAGE MANAGEMENT

* fdisk

- fdisk is used to check the partitions on a disk.
- The fdisk command can display the partitions & details like file system type.

* sfdisk

- sfdisk utility purpose similar to fdisk, but with more features.
- sudo sfdisk -l

* cfdisk

- cfdisk is a Linux utility purpose similar to fdisk, but with more features.
- cfdisk is a Linux partition editor with an interactive user interface based on ncurses.

* parted

- parted utility is to list out partitions & modify them if needed.
 sudo parted -l

- * `df`
 - `df` is not a partitioning utility, but prints out details about only mounted file systems.
 - `df -h`

* Logical Volume Management (LVM)

Installation cmd
`apt install lvm2`

Verify package installation
`dpkg -L lvm2`

Check utilities version

`pvcreate` -- version
`vgcreate` -- version
`lvcreate` -- version.

10/10