Notes2

What is Virtualization? Replication of hardware to stimulate a virtual machine inside a physical machine is called virtualization. Benefits: It allows to run multiple Oss on one machine. It allows to test applications before installing them to host machine. It reduces cost as there is no need to buy a physical hardware for operating different Oss. It allows to experiment with untested program, without infecting host machine. It is hard for a hacker to hack it and so the data remains safer.

Virtual box It runs on: Windows Linux Macintosh Solaris It supports many guest operating systems. After installing virtual machine how to prepare it?

```
Steps: Name your machine and operating system.
```

Add Memory size. Create Hard disk. Select hard disk file type. Select storage from physical hard disk. Setup file location and size Once done with this step, and your machine is successfully setup, change the machine setting as needed. After setting it up turn on your machine and follow the prompts.

Notes3

Exploring Desktop Environment Desktop Environment (DE) is an implementation of the desktop metaphor made of a bundle of programs running on top of computer operating system, which shares a common GUI, sometimes described as a graphical shell.

On windows and macOS the user is limited to a single GUI and DE. However, with Linux, the GUI choice are overwhelming and flexible.

A DE provides a predetermined look and feel to the GUI.

What is a shell? Shell makes largest-scale IT possible. They are necessary component to modern computing. The bash shell is shipped with almost every computer in the world.

The Linux Terminal: CLI: the command-line interface is a means of interacting with a computer program where the user issues commands to the program in the form of successive lines. There are two ways to access CLI: terminal emulator, Linux console.

The Bash Shell:

The GNU bash shell is a program that provides interactive access to the Linux system. It runs as a regular program and is normally started whenever a user logs in into a terminal. Most Linux distributions use the bash shell as the default shell. However, other shells exist.

The Linux filesystem There are a lot of file managers option for Linux. The filesystem is like a tree where every branch represents a directory (folder). The directory where you are now is called current working directory or present working directory. In a filesystem, every file has a pathname which indicated the location of the files in the filesystem.

```
Commands to move around the file: Pwd – used for displaying current working
directory. Cd – used for changing directory. Ls – to display files inside a
```

```
given directory.

Types of pathname: Absolute path Relative path
```

Notes4

Managing files and directories: The commands are often followed by options and then arguments which are the items upon which the command acts on. Example: Command -option argument ls -l ~/Downloads : it will long list all the files in Downloads directory.

```
Commands: Mkdir: to create directories. Mkdir -p: to create a parent
directory with other directories. Touch: to create files. rm: to remove the
files. rm -r: to remove the directory. Mv: to move and rename the
directories using relative path. Sudo mv: to move and rename directories
using absolute path. Cp: to copy the files, to copy the files -r option
must be used. To display inode data: stat + file To create hard link: ln
file ~/Downloads/fileHL Soft link: ln -s files fileSL For getting help: man
ls, to exit man page ‘q’.
```

Wildcards: Star (*) wildcard: it matches anything and nothing and matches any number of characters. Example: ls *.txt – it will list all the files who has .txt as an extension, regardless the name and size of the file.

When to use * wildcard: To list all the files with a particular file extension When you do not know the complete name of file, but you remember the portion of the name. When you want to copy, move or remove all files that match with a particular name convention.

Example: ls *.txt .pdf – to list all files with the extension .txt and .pdf ls file. - to match the name of the file regardless the extension ls file. - to list the file that has string ‘file’ in its name.

Question mark (?) wildcard: it matches precisely one character. It is very helpful when you want to list the hidden files. Example: ls .??* - it will match all the files that start with a or ..and have any character after it.

Bracket [] wildcard: it matches a single character in range. It uses ! mark to reverse the match or say to make an exception. Example: match everything except vowels [!aeiou] or except numbers [!0-9].

For example: Ls f[aeiou]* : to match all files that have a vowel after letter f. Ls f[a-z]* : to match all the files that has range of letter after f

To list the files who has 1990-2000 in its name Ls 1990..2000

Brace {} Expansion: its not a wildcard but another feature of bash that allows you to generate arbitrary strings to use with commands. To create a whole directory structure In a single command: Mkdir -p music/{jazz,rock}/{mp3files,video,oggfiles}/new{1..3}

Parent directory: music Sub directory: jazz – contains: mp3files: new, old, video: new, old, oggfiles: new, old. Rock - contains: mp3files: new, old, video: new, old, oggfiles: new, old. To remove multiple files in single directory: Rm -r {all the files you want to remove}.

Notes5

The basics of Vim: The vi command-line editor is included in all PSOIX complaint OS. Vi has evolved into many different forms including vim, which stands for vi improved. Vi and vim command are separate from each other. To install vim: sudo apt install vim. To start vim, give command on the terminal: vim.

```
Commands: i: to insert text. : prefix for entering command line mode. q:
short for quit. a: short for all buffers ! force :qa! Quit all now.

To create a file and open vim at the same time: [vim + notes.txt]. Use
arrow keys to move around. Enter key to continue in new line. Backspace for
deleting.

Delete text and copy and paste: Dw: delete current word. U: undo. Dd:
delete line under the cursor. D +/word: delete until the word given. Yw:
copy the current word. P: paste after or before the cursor. Yy: copies a
whole line. X: for cut.
```

Notes6

Managing data:

```
Archiving utilities: Tar: creates archive by combining files and
directories into a single file. CPIO: creates an archive, restores files
from an archive, or copies directory. Ar: creates, modifies, and extract
from archive.

The tar program: To create an archive: tar + option (-cf) + archive name +
files to add to archive. To extract an archive: tar + option (-xf) + file
to extract. To extract archive in different directory: tar -xf example.tar
-directory ~/Downloads. Extract specific file: tar -xf example.tar file3.
List the contents of an archive: tar -tf example.tar Add file to archive:
tar -rf example.tar file4. Update file inside an archive: tar -uf
example.tar file4. To add members of an archive to another archive: tar -Af
example.tar example2.tar. To delete specific member of an archive: tar -
delete -f example.tar file3. To compare files with members of an archive:
tar -df example.tar file2.
```

The cpio program: Cpio requires a list of file to archive and option to create an archive is -o. [ls | cpio -ov > archive.cpio]. To extract an archive: cpio -iv < archive.cpio. Archive specific files: find . -iname *.sh | cpio -ov > scriptsarchive.cpio. Create tar archive with cpio: ls | cpio -ov -H tar -F sample.tar. Extract tar archive with cpio: cpio -idv -F smple.tar. View the content of *.tar Archive file: cpio -it -F sample.tar.

```
The ar utility: Archive files with ar: ar r test.a *.txt. List content: ar
t test.a. Add new member to an archive: ar r test.a test3.txt. Delete a
member: ar d test.a test3.txt.

File Compression: The gzip, bzip2, and xz commands are used for
```

```
compression. When you compress file using this command, the result is a
file with similar name but with the correspondent file extension.

Gzip: Compress a single file: gzip file.txt. Multiple files: gzip + file
names To compress and keep original file: gzip -k file.txt Decompress: gzip
-d file.txt Force compression: gzip -f file.txt See details of compressed
file: gzip -l file.txt. Compress files recursively: gzip -r schoolfiles
Test the validity of compress file: gzip -t files.txt.gz. Compress a file
to its max: gzip -9 files.txt.gz. Compress to its min: gzip -1 file.txt.gz

Bzip2: Compress a file: bzip2 file.txt Compress multiple files: bzip2 +
files to compress. Decompress file: bzip2 -d file to decompress. Compress
and keep the file: bzip2 -k + file to decompress.
```

Compress file and show detail: bzip2 -v + files. Check integrity of a file: bzip2 -t file.txt.bz2.

```
Xz: Compress a file: xz file.txt Compress multiple files: xz + files to
compress. Decompress file: xz -d file to decompress. Compress and keep the
file: xz -k + file to decompress. List compression information: xz -l -v +
files. Compress to its max: xz -9 files.txt. Compress to its min: xz -0
files.txt Check integrity of a file: xz -t file.txt.xz.

7zip: Create and archive: 7z a files.tz files.iso. Extract an archive: 7z e
file.7z. Create and archive with different archive format: 7z a -tzip
file.zip fileExample.iso. See files in an archive: 7z l file.7z. Test
integrity: 7z t file.7z. To archive with password protection: 7za a -p
{password_here} files.7z.
```

rar: Create and archive: rar a archive.rar file1 file2 file3. Extract an archive: unrar archive.rar.

File Permission: (File Ownership) A file can be owned by one user and one group.

Ls -l show s you file user, owner, and group owner. /etc/passwd file contains list of all the users in Linux. /etc/group file contains a list of all the groups in Linux. The Chown command is used for changing group owner. Symbolic Notation:

category Operator Permissions U (user) + (add to existing permissions) R (read) G (group) -(remove from existing permissions) W (write) O (other) = (assign absolute permissions) X (execute) A (all) One of the preceding operators One or more of the preceding permissions

Examples: Chmod u+x script.sh Chmod o-x script.sh Chmod u=rex, g=rw, o=r script.sh Numeric Notation:

Permission Value Read 4 Write 2 Execute 1

Example: Chmod 777 script.sh – it will give a file all the permission that is read write and execute. Chmod 700 script.sh – it will give user all the permissions, and no permission for owner or group.

Chmod 555 script.sh – it gives read and execute permission to all 3: user, owner, and group.