Introduction to Linux & Terminal commands

- * Shell > The shell is the Sinux command line interface, interpreter that provides a command line interface, which interpret our commands and tell the OS what to do. e.g. bash (Bourne Again SHell),
 - * Terminal Emulators > Terminal Emulator is a program that will let us use the terminal im a graphical way.
 - * Command Prompt -> It is a part of terminal where we wate commands.

Commanda

- # pwd -> Print name of current working directory
- # 18 -> List directory contents

 1s -a -> list all the files and subdirectories

 of current working directory exen hidden

 one also.
 - ds -l → Displays the current directory contents in long format.
- # cd -> changes the working directory to your home directory
 - cd.. -> more one back from the current working directory.



cd - > changes the working directory to the

previous working directory.

15 - R > list directory tree recursively

** cat - Concatenate files

cat <file-name> -> Display the content of file

cat > \(\text{file-name} - \rightarrow \) To create a file

cat <\(\text{file-1} \rightarrow \) \(\text{file-2} \rightarrow \) \(\text{file-3} \rightarrow \) \(\text{file-1} \rightarrow \) \(\text{file-1} \rightarrow \) \(\text{file-2} \rightarrow \) \(\text{file-1} \rightarrow \) \(\text{file-2} \rightarrow \) \(\text{file-1} \rightarrow \) \(\text{file-1} \rightarrow \) \(\text{file-2} \rightarrow \) \(\text{file-1} \rightarrow \) \(\text{file-2} \rightarrow \) \(\text{file-2} \rightarrow \) \(\text{file-2} \rightarrow \) \(\text{file-1} \rightarrow \) \(\text{file-1} \rightarrow \) \(\text{file-2} \rightarrow \)

* echo -> display a line of text.

echo "Hello world" -> It display Hello world

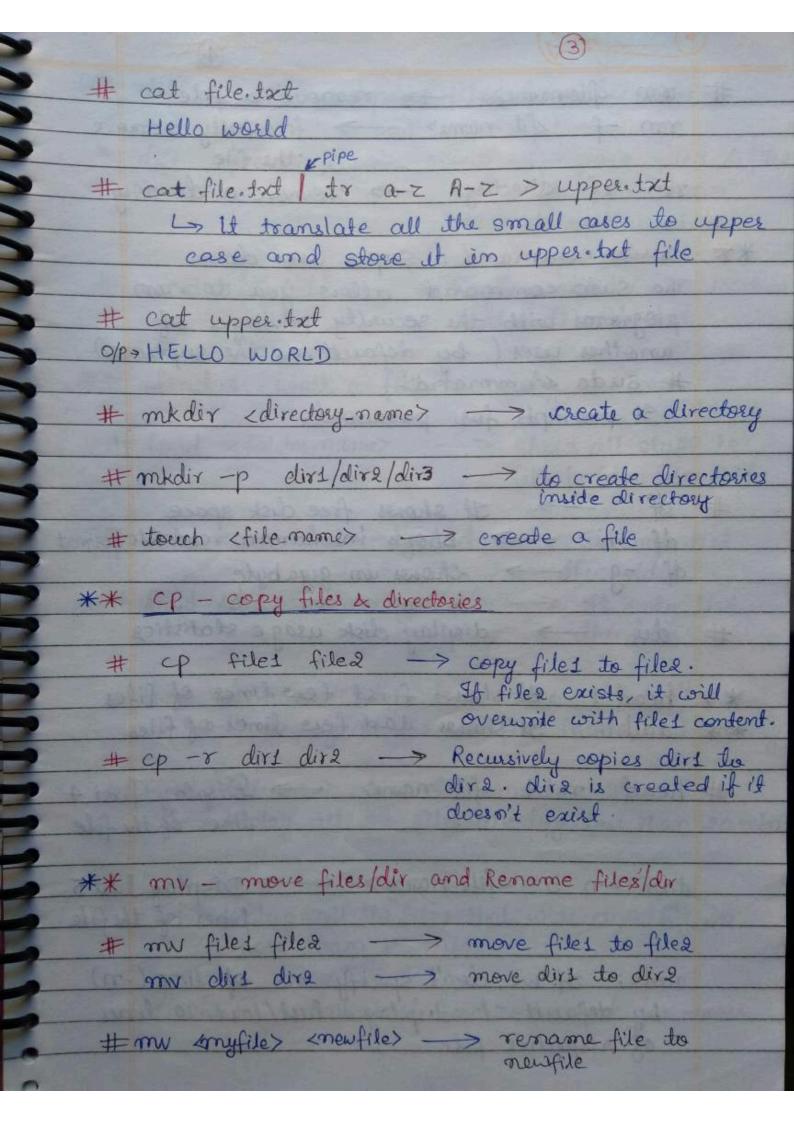
* man [command] -> Display help pages (manual)
of <command>.

** Pipe

pipe operator 1

Using pipe, the output of one command can be piped into the input of another.

command1 | command2

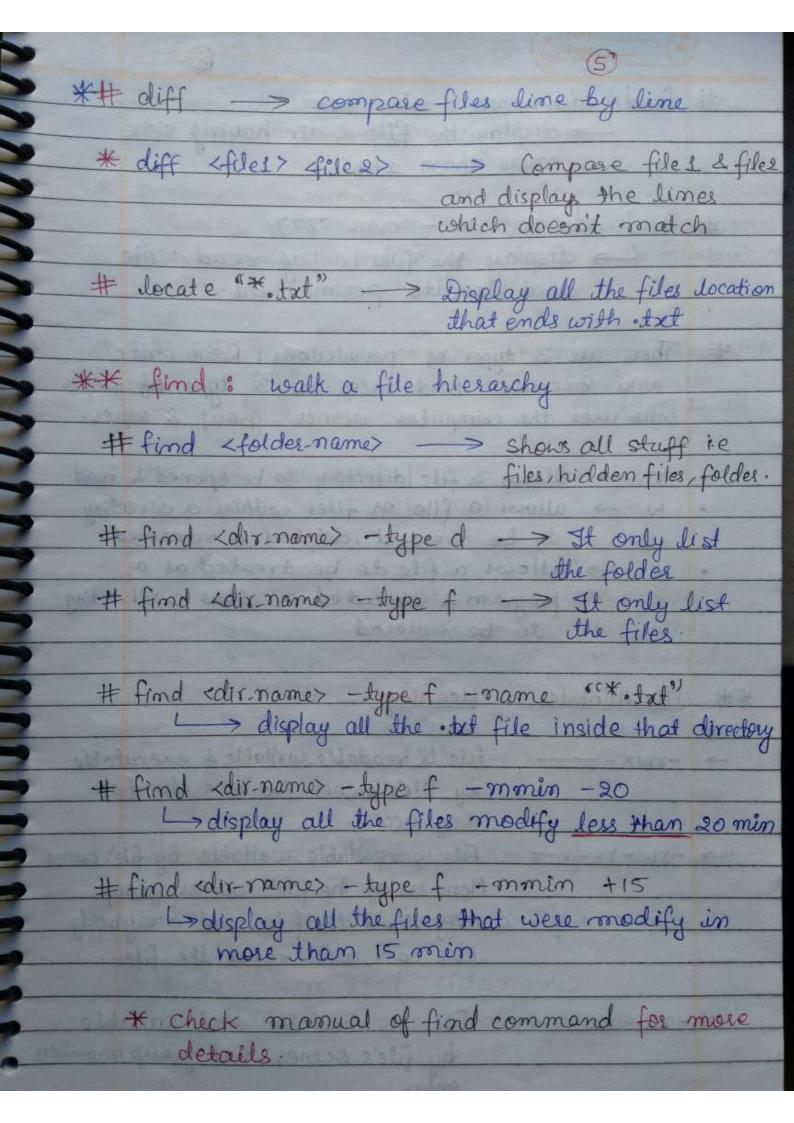


8m file-name> -> remove a file

7m -f \(\) \(\) file-name> -> \(\) the file rm -rf edirectory> -> remove a directory sudo command: super user do The sudo command allow you to run programs with the security privileges of another uses (by default, as the superuses). # sudo «command> It prompts for password. # of -> It shows free disk space

of -h -> shows in human-readable format

of -g -> shows in giga byte # du -> display disk usage statistics It shows first few lines of files
It shows last few lines of files # head -n 4 <file-name> > Digrlay first 4 tail -n 5 <file-name> > Display last 5 lines of the file. note: - If we don't specify no of line (-n)
by default it display first/last 10 lines of respective file.



6

find cdir-name> -size +1k

L> display the files 2 dir having size

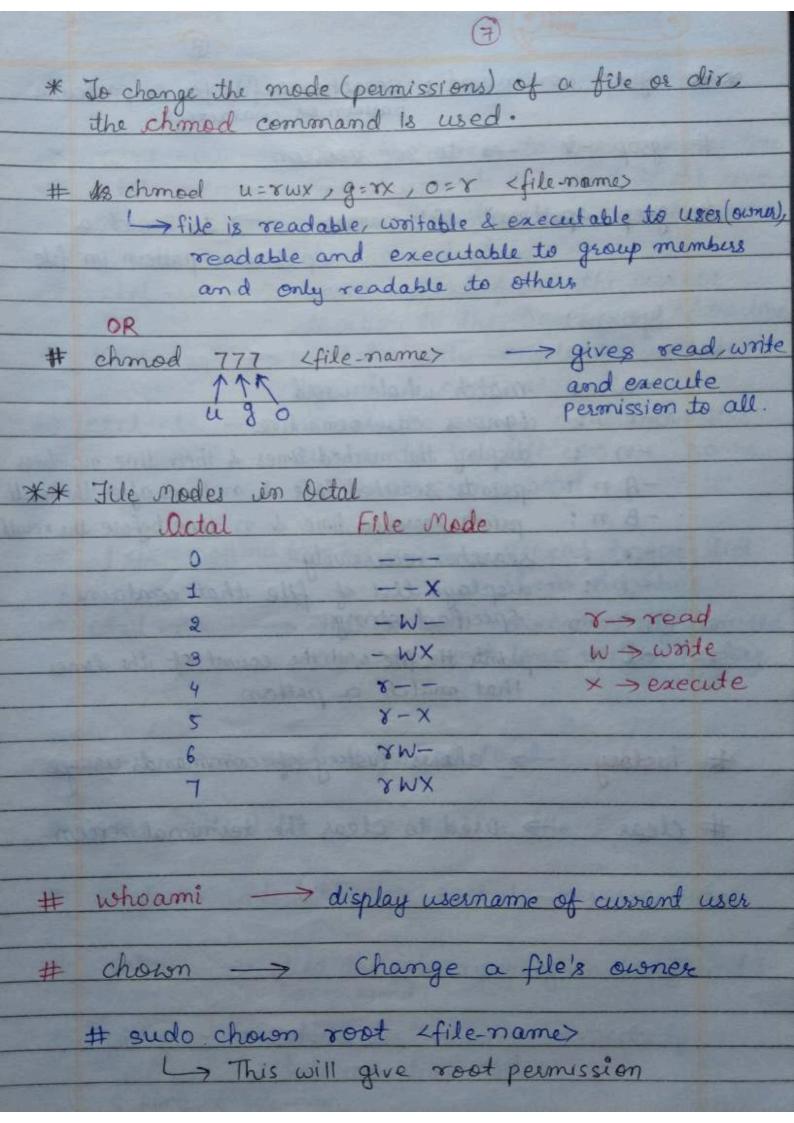
more than 1kb

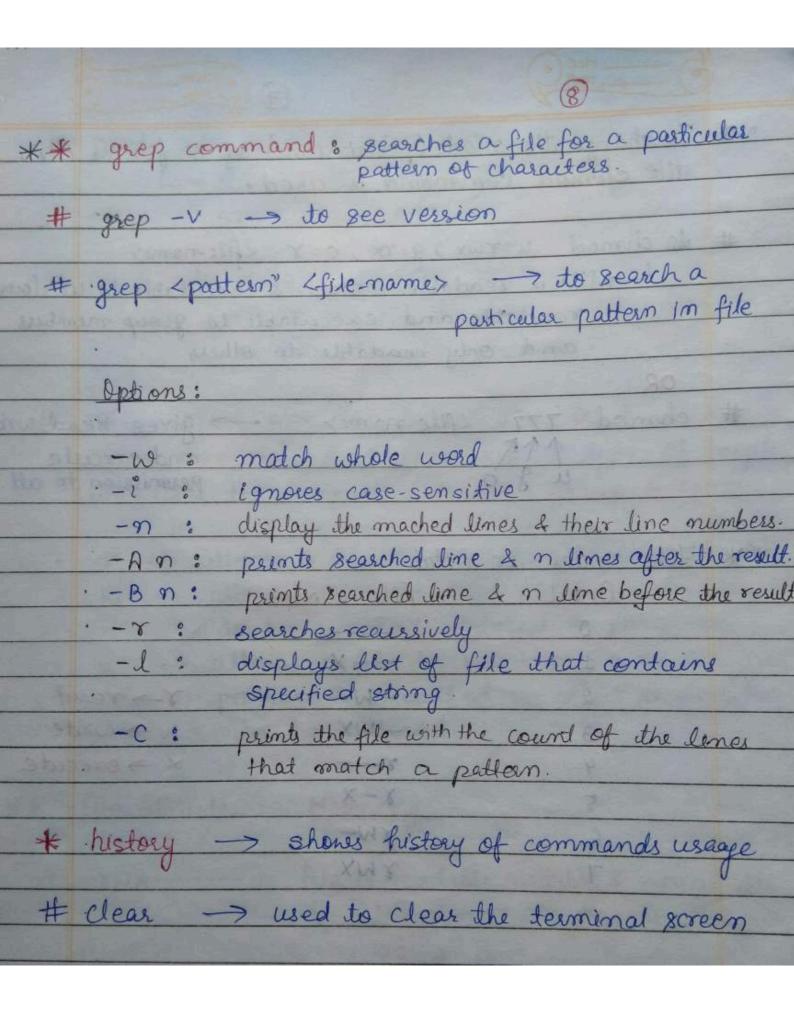
find <dir-name> - perm 777

L> display the files having read write

and execute permissions.

- There are 3 types of permissions: Read, write and Execute. And there are 3 type of people who uses the computer, owner, group & other.
 - · ~ allows a file directory to be opened & read.
 - · W → allows a file or files within a directory to be created , deleted & renamed.
 - · x -> allows a file to be treated as a program & executed or allow a directory to be entered.
- ** File Attributes Meaning
 - by file's owner. No one else has any access.
 - > -8W-8--8-- File is readable, writable by file's owner Members of the file's owner group may read the file and everybody else can also read the file.
 - >-vw-vw--- file is readable and writable by file's owner and group members only.





** Cursor Movement -> Move the cursor to beginning of line ctrl -a \Rightarrow > Move cursor to the end of the line. -> remove text from the cursor location to the end of line. > remove text from the cursos ctol -u docation to the beginning of the line "tab key" -> used for auto completion while typing a command -> clear the screen and move the ctrl -1 cursor to top-left coones some as clear command. => ! < comman-no-from-history -> repeat history list item number => ctrl-r -> This search incrementally from the current and line up the history * Sort -> sort lines of text # sort <file.1xt> -> sort a file # sort -r file. txt -> sort in reverse order -> list the jobs running in background & foreground. # jobs -> checks for network connectivity # pima between two modes. \$ ping google.com

⇒ we can also use IP address to ping directly.

*# free -> display free and used memory -> display the result in megabytes free -m free -h -> display in human-readable format. # vmstat -> to see stat of virtual memory # getent -> checks if a user exist display all the open files Jsof -> # nslookup -> display IP address of the domain # netstat -> display network related info such as routing table, interface statics, etc # cut > cut out selected postions of each line of a file # cut -c 1-3 <file-txt> > print first three characters of each line from # ps aux -> shows processes running status ** "&" operator -> used to combine the commands # ping google.com & ping facebook.com → The function of '&' is to make the command run in background. The command following "&2" operator will execute only if the ** " & &" operator

command preceding this operator has

been successfully executed,

Example # echo "first" & & echo "second" O/P > First second ** OR operator (11) Execute second command only if execution of first command fails. This operator is like an "else" statement in programming. ** NOT operator (1) This is much like an 'except' statement. This will execute all except the condition provides # ron -r ! (*.html) > delate all the files except 'html' files