**Practical-2**

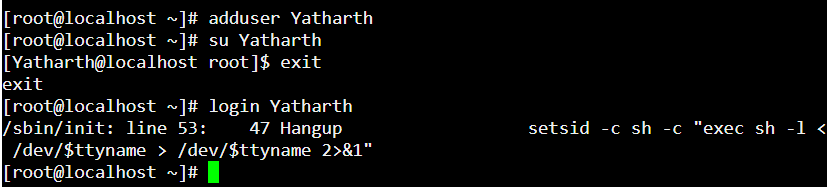
**Aim: Study of Unix Architecture and the following Unix commands with option:**

# User Access:

**Login:** The login program is used to establish a new session with the system. It is normally invoked automatically by responding to the "login:" prompt on the user's terminal.

**Syntax:** login [-p] [-h *host*] [*username*] [*ENV*=*VAR*...]

## Output:

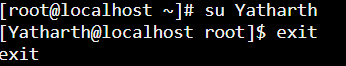
****

**Exit:** When you run exit, if you have jobs running in the background, the shell will remind you that they are running and return you to the command prompt. In this case, issuing exit again will terminate those jobs and exit the shell.

Common aliases for **exit** include "**bye**", "**logout**", and "**lo**".

**Systax:** exit

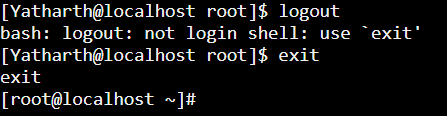
## Output:

****

**logout:** Exits a login shell with exit status N. Returns an error if not executed in a login shell.

**Syntax**: logout [n] Exit a login shell.

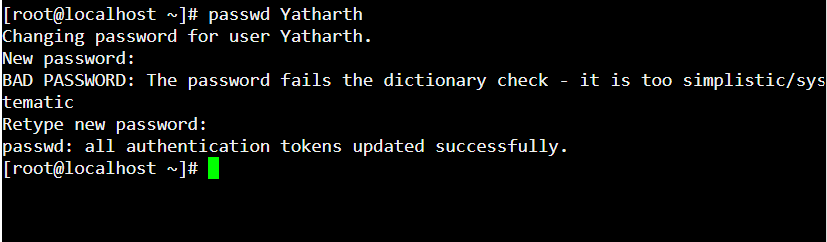
## Output:

****

**passwd:** The passwd command changes passwords for user accounts. A normal user may only change the password for their own account, while the super user may change the password for any account.passwd also changes the account or associated password validity period.

**Syntax**: passwd

**Output:**

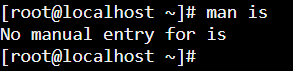
****

# Help:

**Man:** man, command in Linux is used to display the user manual of any command that we can run on the terminal.

**Syntax:** man is

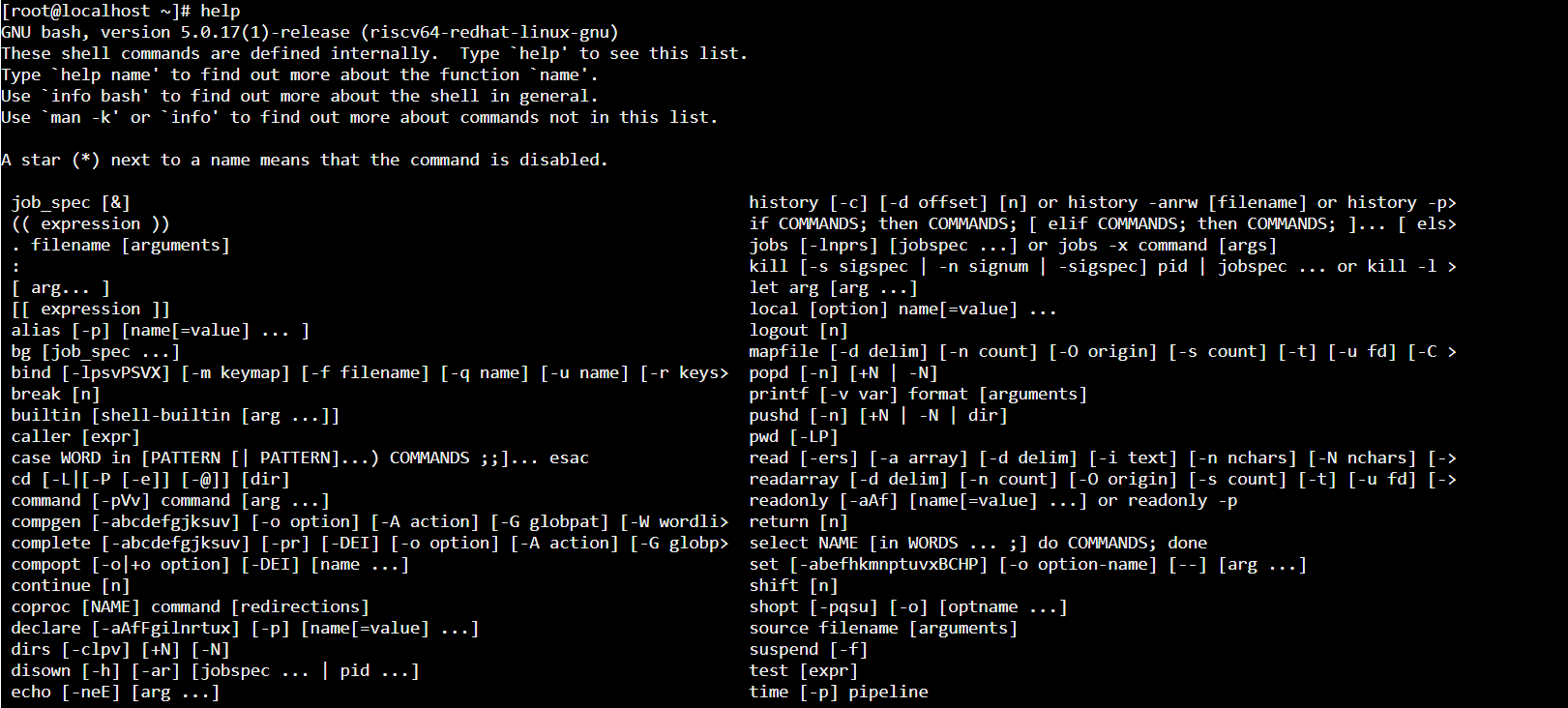
**Output:**

**he**

**Help:**The help command is a shell built-in internal command. It accepts a text string as the command line argument and searches the supplied string in the shell's documents.

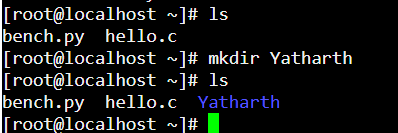
**Syntax:** help [-dms] [pattern ...]

**Output:**

****

# Directory:

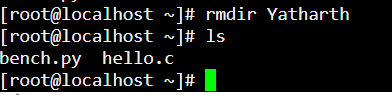
**mkdir**: make directories **Syntax:** mkdir <name> **Output:**



**rmdir:** rmdir command is used remove empty directories from the filesystem in Linux

**Syntax:** rmdir <name>

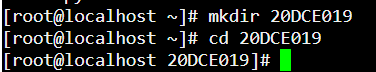
**Output:**

****

**cd :** To change directory - change the current working directory to a specific Folder.

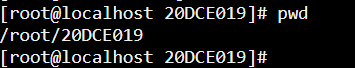
**Syntax:** cd <name>

## Output:

****

**pwd**: pwd stands for Print Working Directory. It prints the path of the working directory, starting from the root.

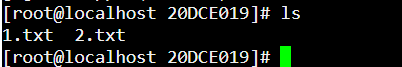
**Syntax: *pwd -L* Output:**



**ls:** ls List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified. Mandatory arguments to long options are mandatory for short options too.

**Syntax:** ls [OPTION]... [FILE]...

## Output:

****

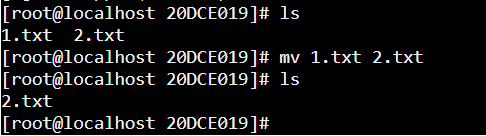
**mv :** mv stands for move. mv is used to move one or more files or directories from one place to another in a file system like UNIX. It has two distinct functions:

1. It renames a file or folder.
2. It moves a group of files to a different directory.

No additional space is consumed on a disk during renaming. This command normally works silently means no prompt for confirmation.

**Syntax:** mv <name><name>

**Output:**

****

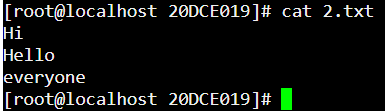
# Editor:

### **vi:** Vim is a text editor that is upwards compatible to Vi. It can be used to edit all kinds of plain text. It is especially useful for editing programs. There are a lot of enhancements above Vi: multi level undo, multi windows and buffers, syntax highlighting, command line editing, filename completion, on-line help, visual selection, etc..

**Syntax:** vim [options] [file ..]

**Output:**

****

****

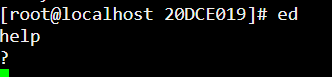
**gedit:** gedit is the default text editor of the GNOME desktop environment and part of the GNOME Core Applications. Designed as a general-purpose text editor, gedit emphasizes simplicity and ease of use, with a clean and simple GUI, according to the philosophy of the GNOME project.

**Syntax:** gedit filename

### **ed:** ed is a line-oriented text editor. It is used to create, display, modify and otherwise manipulate text files. red is a restricted ed: it can only edit files in the current directory and cannot execute shell commands.

**Syntax:** ed [-] [-Gs] [-p *string*] [*file*]

**Output:**

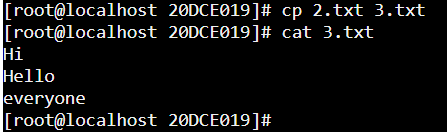
****

# File Handling / Text Processing:

**cp:** To copy one or more files to another location. cp stands for copy. This command is used to copy files or group of files or directory. It creates an exact image of a file on a disk with different file name. *cp* command require at least two filenames in its arguments.

**Syntax:** cp -r

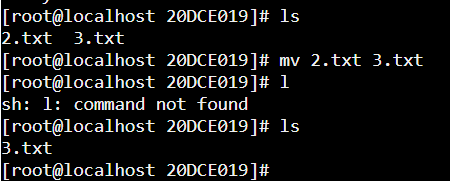
## Output:

****

**mv**: Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY. Mandatory arguments to long options are mandatory for short options too.

**Syntax:** mv [options] [-T] source destination

## Output:

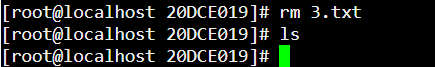
****

**rm:**rm removes each specified file. By default, it does not remove directories. If the *-I* or *-- interactive=once* option is given, and there are more than three files or the *-r*, *-R*, or *--recursive* are given, then rm prompts the user for whether to proceed with the entire operation. If the response is not affirmative, the entire command is aborted.

Otherwise, if a file is unwritable, standard input is a terminal, and the *-f* or *-force* option is not given, or the *-i* or *--interactive=always* option is given, **rm** prompts the user for whether to remove the file. If the response is not affirmative, the file is skipped.

**Syntax:** rm [OPTION]... FILE...

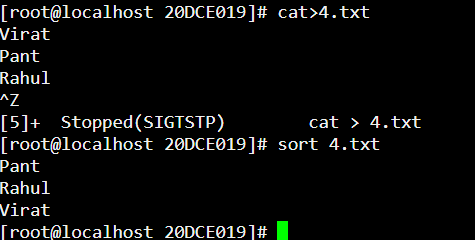
## Output:

****

**Sort:** The sort command is used to sort files in alphabetical order.

**Syntax:** sort <file name>

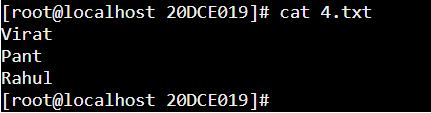
## Output:

****

**Cat:** The cat command is also used as a filter. To filter a file, it is used inside pipes.

**Syntax:** cat <fileName> | cat or tac | cat or tac |. . .

## Output:

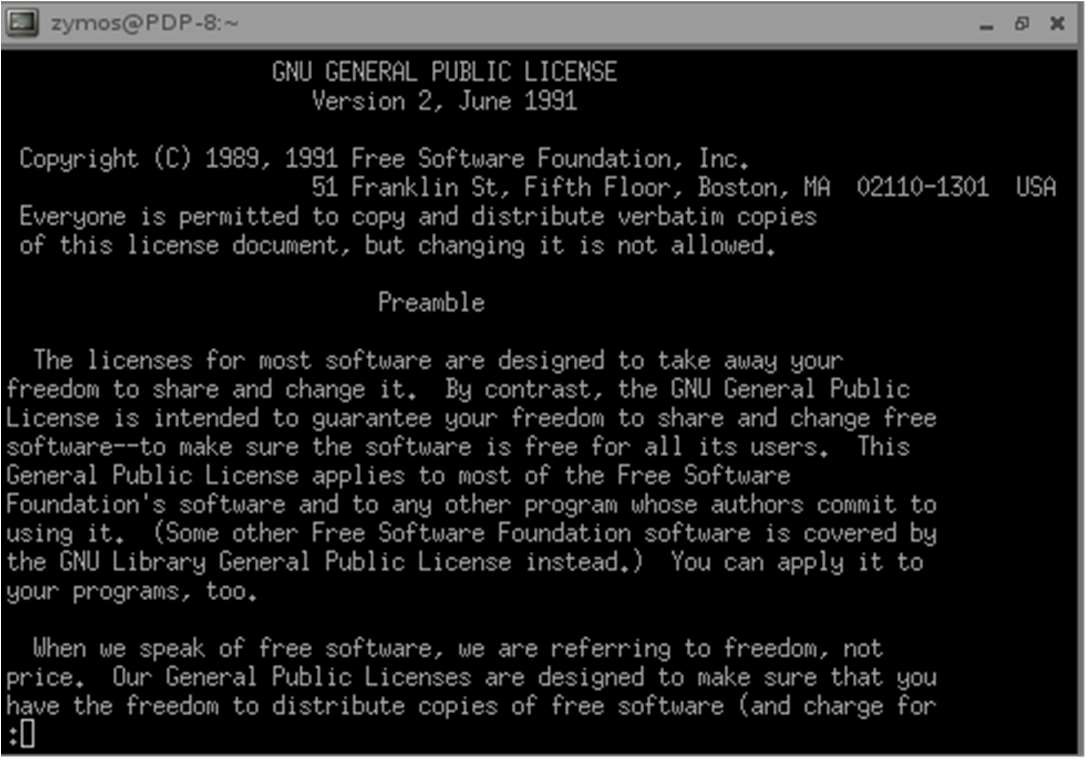
****

**Pg:** pg displays a text file, pausing after each "page" (the height of the terminal screen). After each page, a prompt is displayed. The user may then either press the newline key to view the next page or one of the keys described below.

## Syntax:

pg [-*number*] [-p *string*] [-*cefnrs*] [+*line*] [+/*pattern*/] [*file*...]

## Output:



**lp:** lp submits files for printing, or alters a pending print job. Use a file name of "**-**" to specify printing from the standard input.

## Syntax:

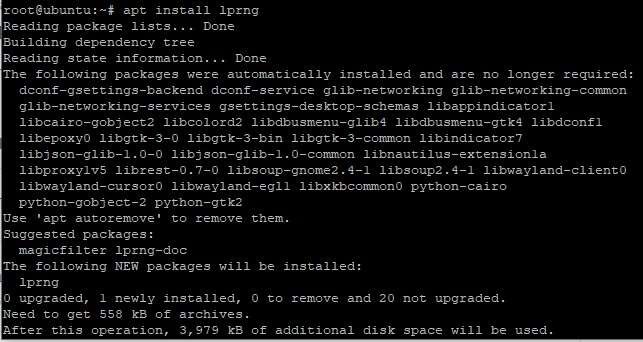
lp -i *request-ID* [-c] [-m] [-p] [-s] [-w] [-d *destination*] [-f *form-name*]

[-H *special-handling*] [-n *number*] [-o *options*] [-p *pagenumbers*]

[-q *priority-level*] [-S *character-set* | *print-wheel*] [-t *title*]

[-T *content-Type* [-r]] [-y *mode-list*]

## Output:

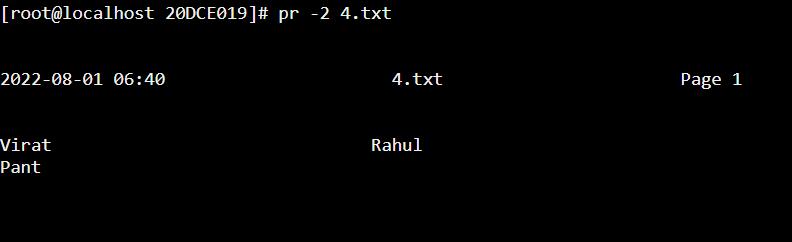


### **pr:** pr paginates or columnates *FILE*(s) for printing.

The **-t** option is implied if PAGE\_LENGTH is less than or equal to **10**. **Syntax:**

pr [*OPTION*]... [*FILE*]...

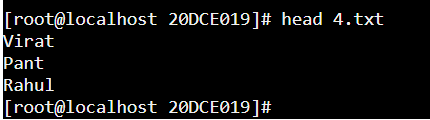
## Output:

****

**head:** The head command is used to display the content of a file. It displays the first 10 lines of a file.

**Syntax:** head <file name>

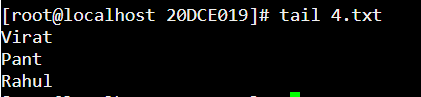
## Output:



**Tail:** It is the complementary of head command.The tail command, as the name implies, print the last N number of data of the given input. By default it prints the last 10 lines of the specified files. If more than one file name is provided then data from each file is precedes by its file name.

**Syntax:** tail [OPTION]... [FILE]...

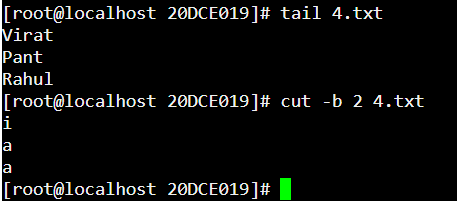
## Output:

****

**cut:** The cut command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and field. Basically the cut command slices a line and extracts the text. It is necessary to specify option with command otherwise it gives error. If more than one file name is provided then data from each file is not precedes by its file name.

**Syntax:** cut OPTION... [FILE]...

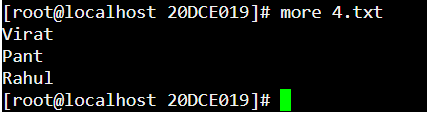
## Output:

****

**more:** The more command is quite similar to the cat command, as it is used to display the file content in the same way that the cat command does. The only difference between both commands is that, in case of larger files, the more command displays screenful output at a time.

**Syntax:** more <file name>

## Output:

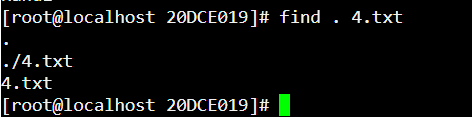
****

**find:** The find command is used to find a particular file within a directory. It also supports various options to find a file such as byname, by type, by date, and more.

The following symbols are used after the find command: (.) : For current directory name

**Syntax:** find. -name "\*.pdf"

## Output:

****

**COMM:** The 'comm' command is used to compare two files or streams. By default, it displays three columns, first displays non-matching items of the first file, second indicates the non- matching item of the second file, and the third column displays the matching items of both files.

**Syntax:** comm <file1> <file2>

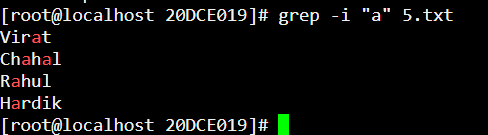
## Output:

****

**Grep**: The grep is the most powerful and used filter in a Linux system. The 'grep' stands for "global regular expression print." It is useful for searching the content from a file. Generally, it is used with the pipe.

**Syntax:** grep <searchWord>

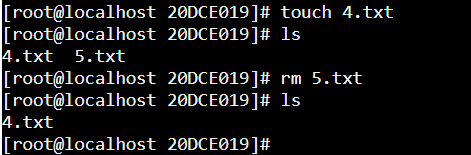
## Output:

****

**Touch:** Update the access and modification times of each FILE to the current time. Mandatory arguments to long options are mandatory for short options too.

**Syntax:**Touch <file.name>

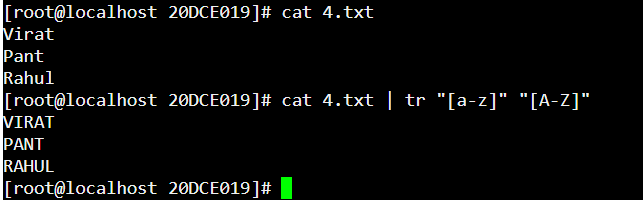
## Output:



**Tr:**The tr command is used to translate the file content like from lower case to upper case.

**Syntax:** <'old'> <'new'>

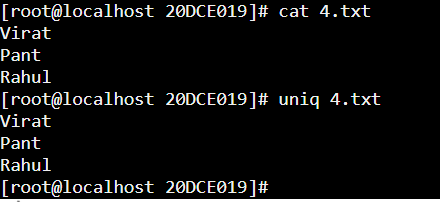
## Output:

****

**Uniq:**The uniq command is used to form a sorted list in which every word will occur only once.

**Syntax:**command <fileName> | uniq

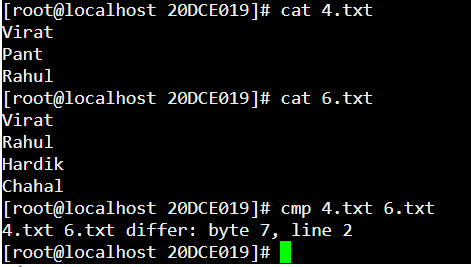
## Output:

****

**Cmp:**'cmp' reports the differences between two files character by character, instead of line by line. As a result, it is more useful than 'diff' for comparing binary files. For text files, 'cmp' is useful mainly when you want to know only whether two files are identical. For files that are identical, 'cmp' produces no output.

**Syntax:**$ cmp sample.txt sample1.txt

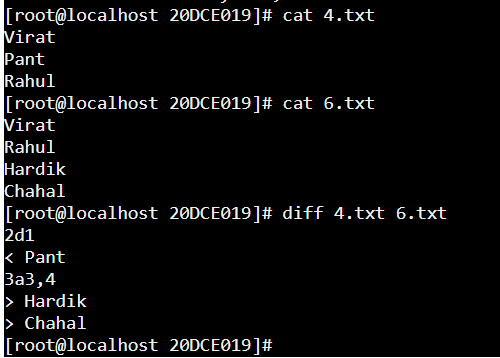
## Output:

****

**Diff:**The diff software does not actually change the files it compares. However, it can optionally generate a script (if the -e option is specified) for the program ed or ex which can be used to apply the changes.

**Syntax:**diff file1.txt file2.txt

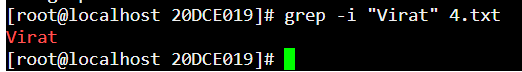
## Output:

****

**Grep:** The grep is the most powerful and used filter in a Linux system. The 'grep' stands for "global regular expression print." It is useful for searching the content from a file. Generally, it is used with the pipe.

**Syntax:** grep <searchWord>

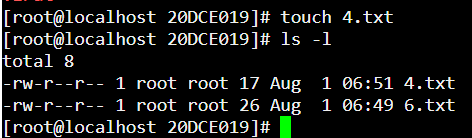
## Output:

****

**Touch:** Update the access and modification times of each FILE to the current time. Mandatory arguments to long options are mandatory for short options too.

**Syntax:** Touch <file.name>

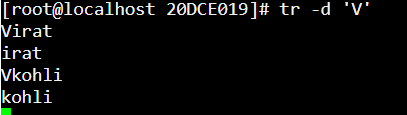
## Output:

****

**Tr:** The tr command is used to translate the file content like from lower case to upper case.

**Syntax:** tr <'old'> <'new'>

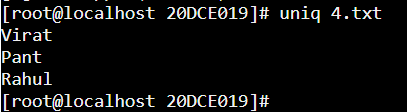
## Output:

****

**Uniq:** The uniq is used to form a sorted list in which every word will occur only once.

**Syntax:** command <fileName> | uniq

## Output:

****

**Security and Protection:**

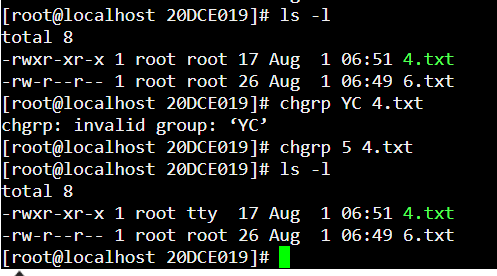
**chmod:** To change modes of a file.

**Syntax:** chmod [reference][operator][mode] file

## Output:

## 

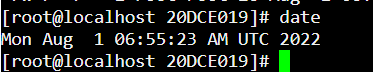
**chown:** To change group of a file. **Syntax:** chgrp [user] [filename.extension] **Output:**



**Date:** This command gives current day, date and time

**Syntax:** date

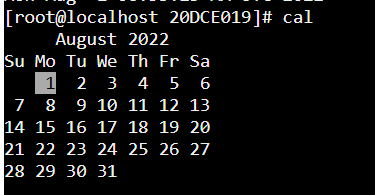
## Output:

****

**cal:** This command returns current month.

**Syntax**: cal

## Output:

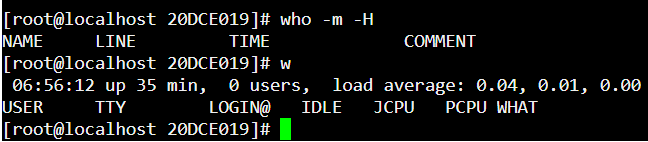
****

**who:** who command is used to find out the following information:

1. Time of last system boot
2. Current run level of the system
3. List of logged in users and more.

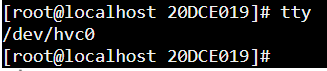
**Syntax:**$who [options] [filename]

## Output:

  
**Tty:** It displays information related to terminal. The tty command of terminal basically prints the file name of the terminal connected to standard input. tty is short of teletype, but popularly known as a terminal it allows you to interact with the system by passing on the data (you input) to the system, and displaying the output produced by the system.

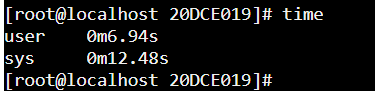
**Syntax:** tty [OPTIONS]

## Output:

****

**time:** time command in Linux is used to execute a command and prints a summary of realtime, user CPU time and system CPU time spent by executing a command when it terminates. ‘*real*‘ time is the time elapsed wall clock time taken by a command to get executed, while ‘*user*‘ and ‘*sys*‘ time are the number of CPU seconds that command uses in user and kernel mode respectively.

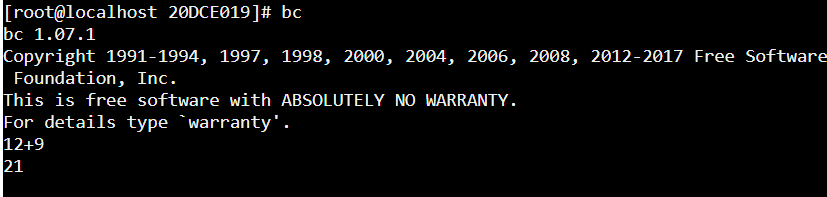
**Syntax:** time [OPTIONS] [COMMAND]



**Bc:** bc stands for Basic Calculations. Bc launches a prompt which allows user to perform basic computations.

**Syntax:** bc

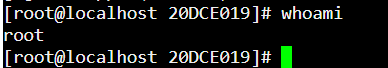
**Output:**

****

**Whoami:** This command returns the current username.

**Syntax:** whoami

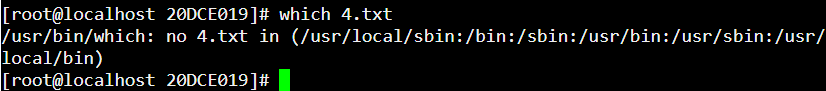
## Output:

****

**Which:** This command returns the whole directory of the command passed as parameter.

**Syntax:** which [COMMAND\_NAME]

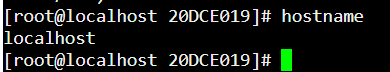
## Output:

****

**Hostname:** This command returns the current hostname.

**Syntax:** hostname

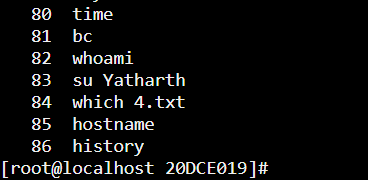
## Output:

****

**history:** history command is used to view the previously executed command. This feature was not available in the Bourne shell. Bash and Korn support this feature in which every command executed is treated as the event and is associated with an event number using which they can be recalled and changed if required. These commands are saved in a history file. In Bash shell history command shows the whole list of the command.

**Syntax:** $ history

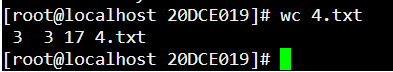
## Output:

****

**Wc:** WC stands for Word Count. This command returns the number of lines, words and letters used in file passed as parameter.

**Syntax:** wc [FILENAME]

**Output:**

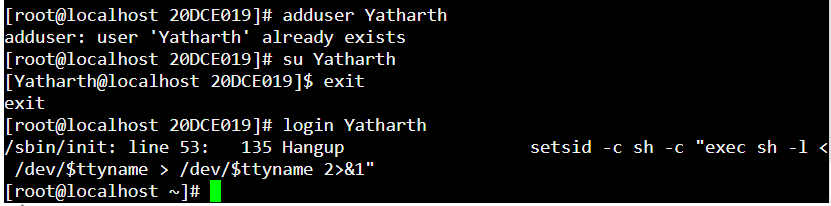
****

# System Administrator:

**Su:** The Unix command su, which stands for 'substitute user', is used by a computer user to execute commands with the privileges of another user account.

**Syntax:** su [USERNAME]

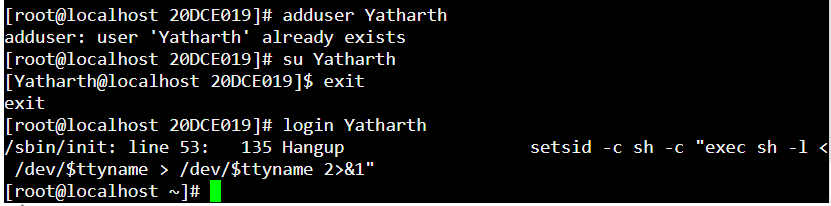
## Output:

****

**Adduser:** This command allows root user to create another user with restricted privilages.

**Syntax:** adduser [username]

## Output:

****

**Rmuser:** This command allows root user to delete another user with restricted privilages.

**Syntax:** rmuser [username]

## Output:

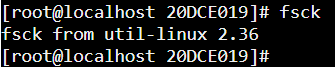


**fsck:** The fsck (File System Consistency Check) Linux utility checks filesystems for errors or outstanding issues. The tool is used to fix potential errors and generate reports.

**Syntax:** fsck <<em>options>

<filesystem></em>

## Output:

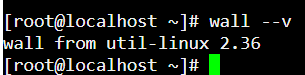
****

**init:** init is the parent of all Linux processes. It is the first process to start when a computer boots up and it runs until the system shuts down. It is the ancestor of all other processes.



**wall:**wall command in Linux system is used to write a message to all users. This command displays a message, or the contents of a file, or otherwise its standard input, on the terminals of all currently logged in users.

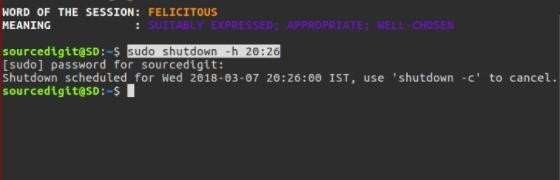
**Syntax:** initwhoma [-n] [-t timeout] [message | file] **Output:**



**shutdown:**The shutdown command in Linux is used to shutdown the system in a safe way. You can shutdown the machine immediately, or schedule a shutdown using 24-hour format. It brings the system down in a secure way.

**Syntax:**shutdown [OPTIONS] [TIME] [MESSAGE]

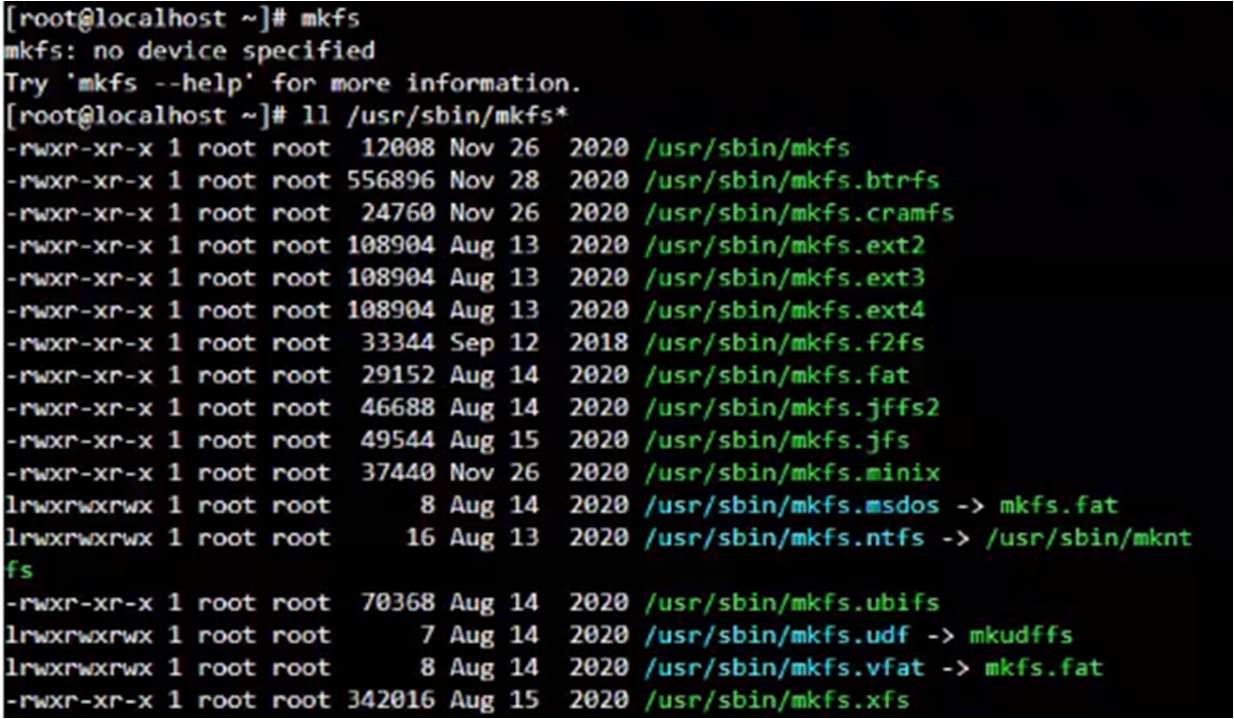
## Output:



**mskfs:** The mkfs command stands for “make file system” is utilized to make a file system (which is, a system for organizing a hierarchy of directories, subdirectories, and files).

**Syntax:** mkfs [ -V ] [ -t fstype ] [ fs-options ] filesys [ blocks ]

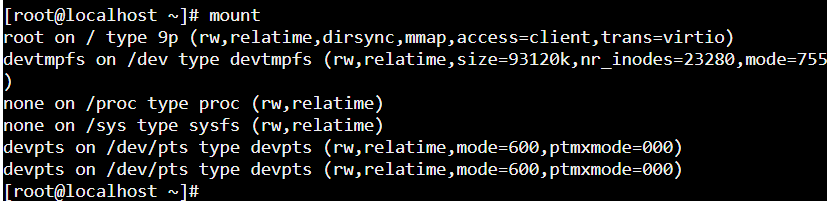
## Output:



**mount:** mount command is used to mount the filesystem found on a device to big tree structure(Linux filesystem) rooted at ‘/‘. Conversely, another command umount can be used to detach these devices from the Tree.

**Syntax:**mount -t type device dir

## Output:

****

**unmount:**This command enables you to remove a remote file system that is currently mounted. The umount command supports the -V option to allow for testing.

**Syntax:** umount [options] filesystem

## Output:



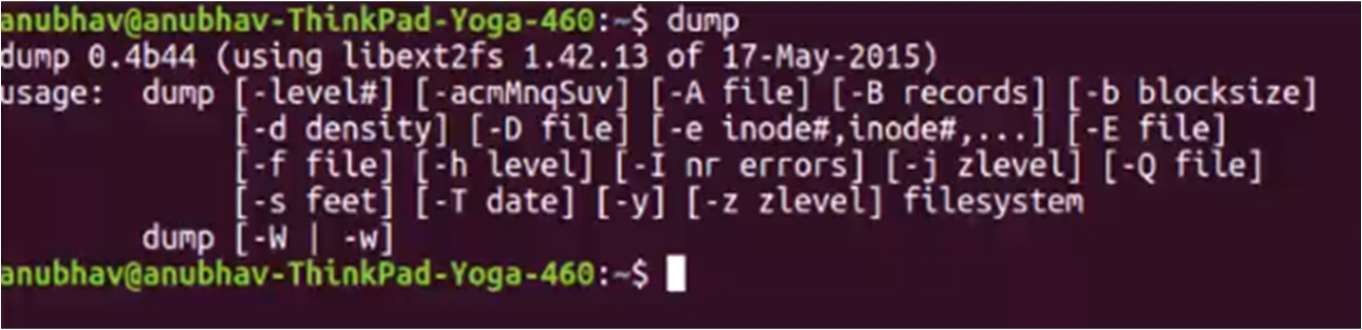
**dump:** dump command in Linux is used for backup the filesystem to some storage device. It backs up the complete file system and not the individual files. In other words, it backups the required files to tape, disk or any other storage device for safe storage.

#### Syntax:

dump [-level#] [-a autosize] [-A file] [-B records] [-b blocksize] [- d density] [-D file] [-e inode numbers] [-E file] [-f file]

[-F script] [-h level] [-I nr errors] [-jcompression level] [-L label] [- Q file] [-s feet] [-T date] [-y] [-zcompression level] files-to-dump dump [-W | -w]

## Output:



**restore:** restore command in Linux system is used for restoring files from a backup created using dump. The restore command performs the exact inverse function of dump. A full backup of a file system is being restored and subsequent incremental backups layered is being kept on top of it.

#### Syntax:

restore -C [-cdHklMvVy] [-b blocksize] [-D filesystem] [-f file] [-F script] [-L limit] [-s fileno] [T directory]

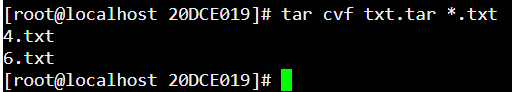
## Output:



**tar:** The Linux ‘tar’ stands for tape archive, is used to create Archive and extract the Archive files. tar command in Linux is one of the important command which provides archiving functionality in Linux. We can use Linux tar command to create compressed or uncompressed Archive files and also maintain and modify them.

**Syntax:** tar [options] [archive-file] [file or directory to be archived]

**Output:**

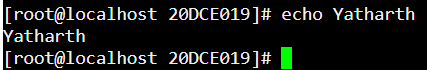
****

# Terminal:

**Echo:** This command allows user to print the string (passed as parameter) with new line character on the console.

**Syntax:** echo [STRING]

## Output:

****

**printf:** This command allows user to print the string (passed as parameter) without new line character on the console.

**Syntax:** printf [STRING]

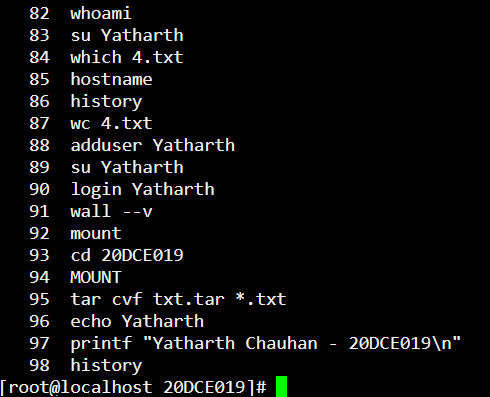
## Output:

****

**Clear:** This command clears the terminal screen and provides blank screen to user.

**Syntax:** clear

## Output:

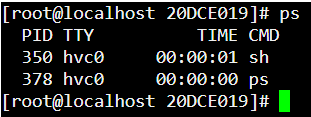
****



**Process:**

**ps:** Linux provides us a utility called **ps** for viewing information related with the processes on a system which stands as abbreviation for **“Process Status”.** ps command is used to list the currently running processes and their PIDs along with some other information depends on different options.

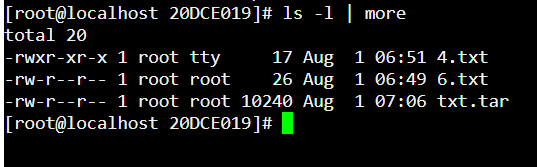
**Syntax: ps [options] Output:**



**Pipe ( | ):** This command executes multiple commands simultaneously.

**Syntax:** [COMMAND1] | [COMMAND2] | …

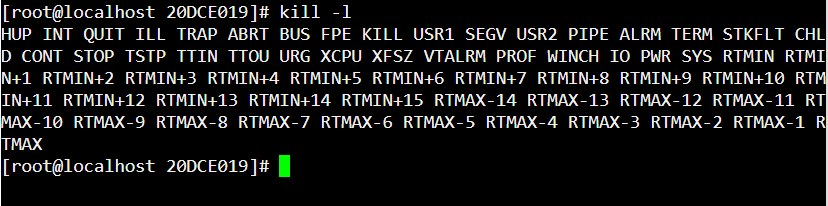
## Output:

****

**kill -l :**To display all the available signals you can use below command option:

**Syntax:** $kill -l

## Output:



**exec**:exec command in Linux is used to execute a command from the bash itself. This command does not create a new process it just replaces the bash with the command to be executed. If the exec command is successful, it does not return to the calling process.

**Syntax:** exec [-cl] [-a name] [command [arguments]] [redirection ...]

**Output:**

****

**I/O Redirection (<,>,>>):** Redirection can be defined as changing the way from where commands read input to where commands sends output. You can redirect input and output of a command.

**Overwrite:** Commands with a single bracket '>' overwrite existing file content.

> : standard output

< : standard input 2> : standard error

**Syntax:** cat > <fileName>

## Output:

## 

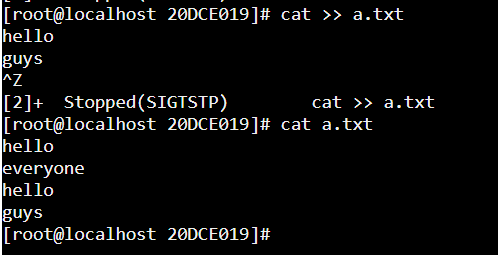
**Append:**Commands with a double bracket '>>' do not overwrite the existing file content.

>> - standard output

<< - standard input 2>> - standard error

**Syntax:**cat >> <fileName>

#### Output:

****

**Gcc:** GCC stands for GNU Compiler Collections which is used to compile mainly C and C++ language. It can also be used to compile Objective C and Objective C++. The most important option required while compiling a source code file is the name of the source program, rest every argument is optional like a warning, debugging, linking libraries, object file etc. The different options of gcc command allow the user to stop the compilation process at different stages.

**Syntax:** gcc [C\_Source] [OPTION] [COMMAND]

**Output:**

