

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

**CSE-202 OPERATING SYSTEM**

Practical File

Submitted By:

- Avichal Suneja

- A2305219437

- 4CSE7X………..

Submitted To:

**Ms Abha Sachdeva**

EXPERIMENT-1

Aim:

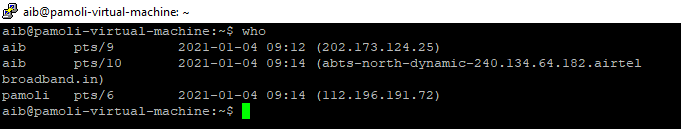
Implement different commands in UNIX

**1) Who:** It displays the name of the machine and time of logins.

Syntax: $who [options]

Example: $ who –m -H

Output:

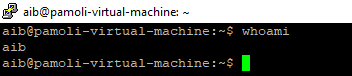


**2) whoami:** Display the account name associated with the current login.

Syntax: $whoami

Example: $whoami

Output:

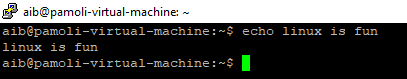


**3) echo:** echo command in UNIX is used to display line of text/string that are ……………passed as an argument.

Syntax: $echo [option] [string]

Example: $echo Linux is fun.

Output:



**4) cal:** It displays the calendar according to the user's requirement.

Syntax: cal [month] [year] [-m month] [-y year] [-h] [-3] [-1] [-A num]

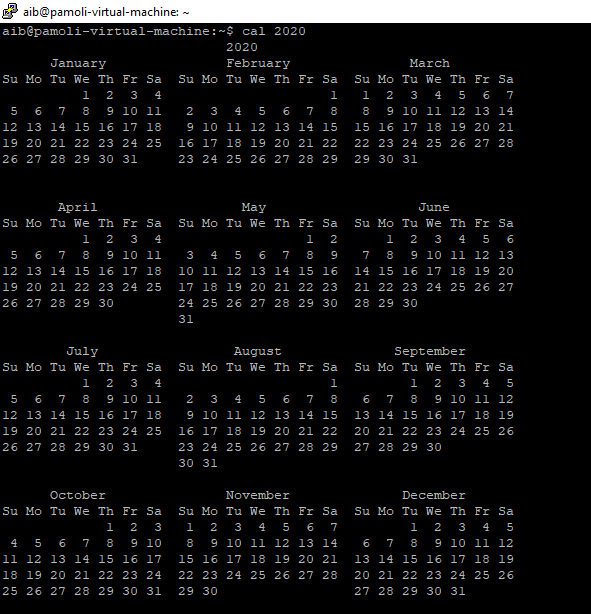
[-B num] [-d YYYY-MM] [-j] [-N] ncal [month] [year] [-m month]

[-y year] [-h] [-3] [-1] [-A num] [-B num] [-d YYYY-MM] [-J] [-C]

[-e] [-o] [-p] [-w] [-M] [-S] [-b]

Examples: $ cal 2020, $cal 2000., $cal -1

Output:

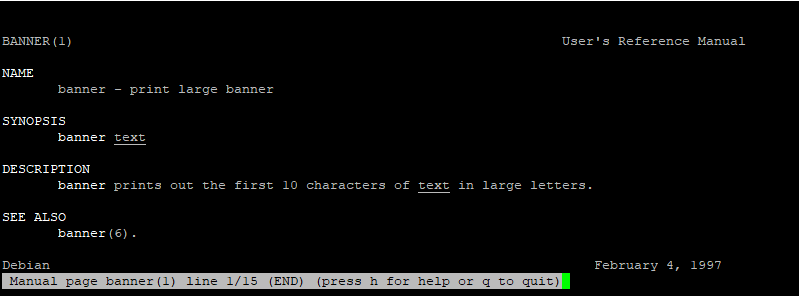


**5) man**: Man command in UNIX is used to display the user manual of any ……………command that we can run on the terminal.

Syntax: $man [COMMAND NAME]

Examples: $man ls, $man banner

Output:

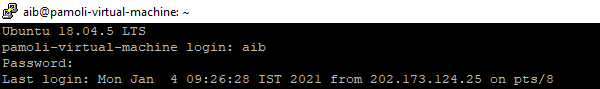


**6) login**: The command is used to login into secure operating system with the …………….help of username and password.

Syntax: $login

Example: $login <marco>

Output:



**7) logout:** It logs out of the operating system.

Syntax: $logout

Example: $logout

Output:



**8) exit**: It exits from the terminal and comebacks to the desktop.

Syntax: $exit [n]

Example: $exit

Output:

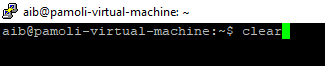


**9) clear:** It clears the terminal window

Syntax: $clear

Example: $clear

Output:

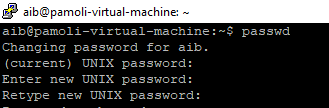


**10) passwd:** It is used to change the user account password

Syntax: $passwd [options] [LOGIN]

Example: $passwd

Output:

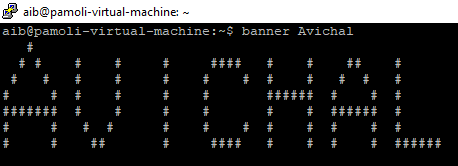


**11) banner:** banner command in linux is used to print the ASCII character string ………………….in large letter to standard output.

Syntax: $banner text

Example: $banner Avichal

Output:



**12) pwd:** pwd command short for print working directory, and it will show the …………….current working directory

Syntax: pwd

Example: $pwd

Output:

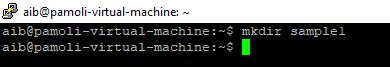


**13) mkdir:** We can create new directory using mkdir command. For example, ………………..we will create a code directory inside our home directory

Syntax: mkdir [FolderName]

Example: $mkdir sample1

Output:

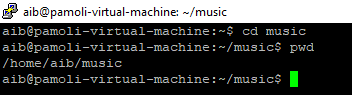


**14) cd:** cd, short for change directory. This command will help you to change …………..your current directory. We will move to the Music directory in our …………..example from home.

Syntax: cd [path]

Example: $cd music

Output:

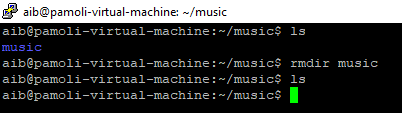


**15) rmdir:** rmdir short for remove directory. This command will help to delete ……………….an empty directory in the present working directory

Syntax: rmdir [dir\_name]

Example: $rmdir music

Output:

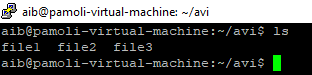


**16) ls:** This command is used to list all the files and directories

Syntax: ls

Example: $ls

Output:

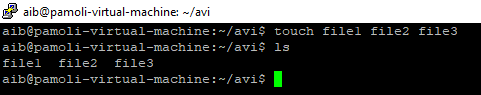


**17) touch:** Used for creating a blank file or a set of multiple blank files

Syntax: touch [File\_Name], [File\_Name], …n

Example: $touch file1 file2 file3

Output:



**18) rm:** Used to remove a file or directory . The -r option is being used to …………..remove in a recursive way. With -f you force the removal ignoring …………..errors and never prompt. You can chain the flags so, instead of rm -r -f …………..you can as well type rm -rf.

Syntax: rm -r [folderName] or [filename}

rm -r -f [folderName] or [filename]

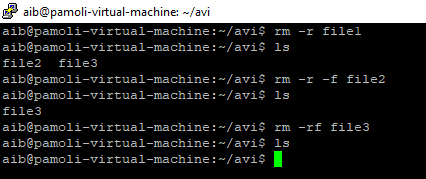
rm -rf [folderName] or[filename]

Example: $rm -r file1

$rm -r -f file2

$rm -rf file3

Output:



**19) cat:** We can create a file with a cat command and also we can view the …………..content of the file with this command.

Syntax: cat > filename (for creating a file)

cat filename. (for viewing content of a file)

cat file1 >> file2 (append content of file 1 into 2 at the end)

cat file1 > file 3 (Creating a new file3)

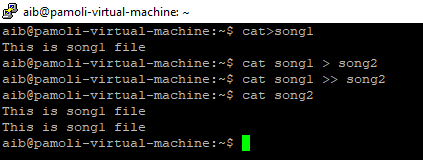
Example: $cat > song1.txt

$cat song1.txt

$cat song1 > song2

$cat song1 >> song2

Output:

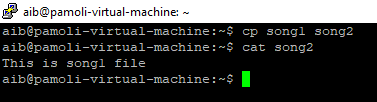


**20) cp:** We use the cp command to copy a file in the Linux shell. To copy a ………….folder with its contents recursively use the cp command with the -r flag.

Syntax: cp [filename] [new filename]

Example: $cp song1 song2

Output:

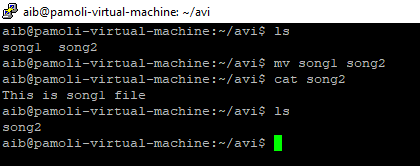


**21) mv:** The mv command is used to rename or move a file or directory.

Syntax: mv [filename] [new filename]

Example: $mv song1 song2

Output:



EXPERIMENT-2

Aim:

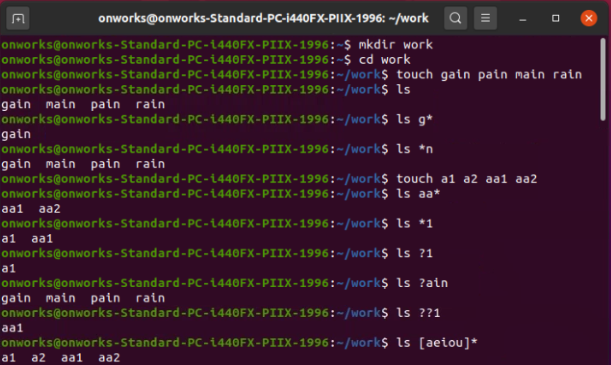
To perform various operations on files in UNIX using various commands.

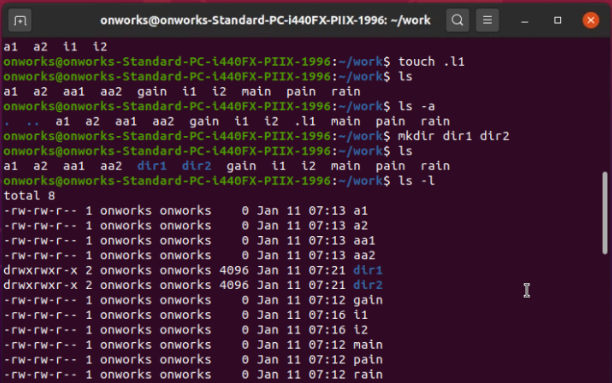
1. **ls:** The ls command is used to list all contents in a folder. It has many flags to list certain files, for example, all files starting from letter ‘a’, etc.

Syntax: ls [options] [paths]

Example: $ls [aeiou]\*

Output:



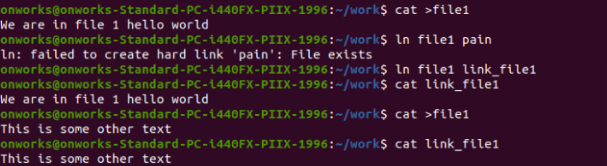


1. **ln:** link command creates a link between an existing file and a new file. All the changes which are made in the original file are automatically displayed in the linked file.

Syntax: ln[option] link name

Example: $ ln file1 link\_file1

Output:

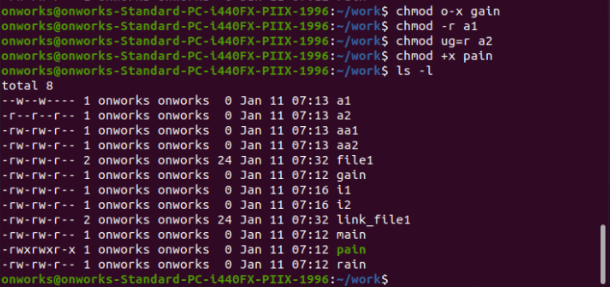


1. **chmod:** chmod or change mode command can be used in UNIX to change the permissions that our files have. It can either add or delete certain permissions like reading and writing from the files.

Syntax: chmod [options][mode] filename

Example: chmod o-x gain

Output:

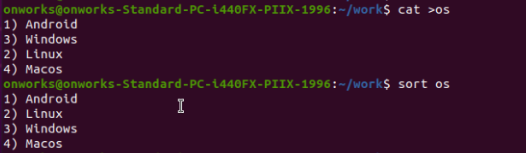


1. **sort:** sort command in UNIX is used for sorting the contents of a file depending upon their numeric ordering or alphabetical order.

Syntax: sort[options] filename

Example: $ sort os

Output:



1. **wc:** wc stands for word count. This command is used in UNIX to get the total word count of a file. It can also display the number of lines and characters with the -l and -c options.

Syntax: wc[options] filename

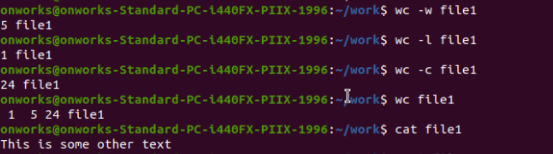
Example: $ wc -w file1

$ wc -l file1

$wc -c file1

$ wc file1

Output:

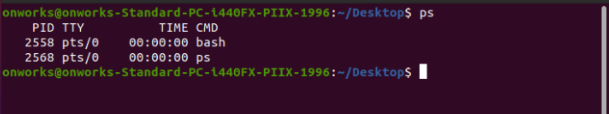


1. **ps:** In its simplest form, when used without any option, ps will print four columns of information for minimum two processes running in the current shell, the shell itself, and the processes that run in the shell when the command was invoked.

Syntax: PS

Example: $ ps

Output:

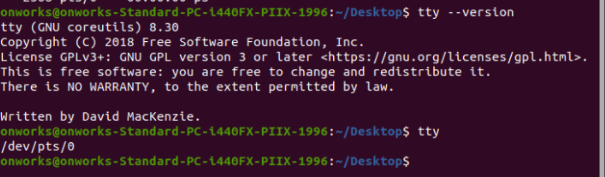


1. **tty:** The tty command of terminal basically prints the file name of the terminal connected to standard input. tty is short of teletype, but popularly known as a terminal it allows you to interact with the system by passing on the data (you input) to the system, and displaying the output produced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

Syntax: tty [Options]

Example: $ tty –version

Output:

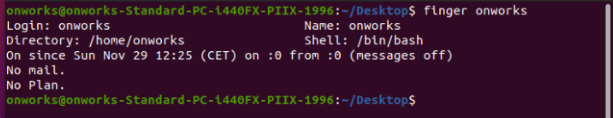


1. **finger:** Finger command is a user information lookup command which gives details of all the users logged in. This tool is generally used by system administrators. It provides details like login name, user name, idle time, login time, and in some cases their email address even.

Syntax: finger username

Example: $ finger onworks

Output:



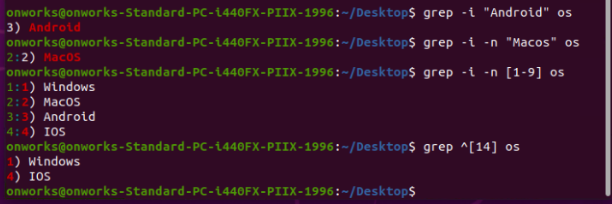
1. **grep:** The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out).

Syntax: grep [options] pattern [filename]

Example: $ grep -I “Android” os

$ grep -I -n [1-9] os

Output:



1. **more:** more command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large (For example log files). The more command also allows the user do scroll up and down through the page. The syntax along with options and command is as follows. Another application of more is to use it with some other command after a pipe. When the output is large, we can use more command to see output one by one.

Syntax: more [-options] [-num] [+/pattern] [+linenum] [file\_name]

Example: $ more -d file

Output:

