

DEPARTMENT OF INFORMATION TECHNOLOGY

3.To execute File Management Commands : LO3

Aim: To execute File Management Commands in unix

Ls : The ls command in Unix (and Unix-like operating systems) is used to list the contents of a directory. By default, it displays the names of files and directories in the current working directory.

```
lab1003@lab1003-OptiPlex-3070:~$ ls
20          demo1      ls1.tcl      's13 51 assignment 5.docx'
34.tcl      demo.bz2    ls2.tcl      $2192
35b.sh      Desktop     ls.tcl      seit
381.tcl     Documents   main.tcl    split_fileaa
382.tcl     Downloads   movie.txt  split_fileab
38.tcl      evencolumns.txt Music      split_fileac
41.pl       examples.desktop mybank.lst string.pl
anesh.lst   example.tcl  myfile.lst student_data.txt
anesh.nam   f1          myfile.lst tabShell
assign7     fib.pl      newa.c      temp
assign8_2   file1.txt   newb.c      Templates
basic.pl    file1.txt.bz2 Newfile.txt 'udo -l [-AknS] [-a type] [-g group] [-h host] [-p prompt] [-U user]'
Basicsi1.pl file5.gz    out.nam    'Untitled Document 1'
calculator  first.c    out.tr     Videos
capture.pcap geekfile.txt Pictures   xaa
capture_tcpd ketan.nam  power.pl   xab
comp.lst    ketan.tr   prime.pl   xac
comp.txt    leapshell  Public     xad
data.lst    lgfile     rectShell
Lab1003@Lab1003-OptiPlex-3070:~$
```

Ls -l :

```
total 16
-rw-rw-r-- 1 lab1003 lab1003 50 Jan 30 16:06 Hello1.sh
-rw-rw-r-- 1 lab1003 lab1003 33 Jan 30 16:01 Hello.sh
-rw-rw-r-- 1 lab1003 lab1003 51 Jan 30 15:28 Name.txt
-rw-rw-r-- 1 lab1003 lab1003 28 Jan 30 15:39 text2.txt
Lab1003@Lab1003-HP-280-G2-MT:~/Pratham_63$ ls -l
```

```
-rw-rw-r-- 1 lab1003 lab1003 22 Feb  6 14:02 Test1.txt
-r----- 1 lab1003 lab1003 28 Jan 30 15:39 text2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ ls -la
total 52
drwxrwxr-x  2 lab1003 lab1003  4096 Feb  6 14:07 .
drwxr-xr-x 27 lab1003 lab1003  4096 Feb  6 13:37 ..
-rw-rw-r--  1 lab1003 lab1003   49 Jan 30 16:09 Hello1.sh
-rw-rw-r--  1 lab1003 lab1003   33 Jan 30 16:01 Hello.sh
-rw-rw-r--  1 lab1003 lab1003 12288 Jan 30 15:58 .Hello.sh.swo
-rw-rw-r--  1 lab1003 lab1003   51 Jan 30 15:28 Name.txt
-rw-rw-r--  1 lab1003 lab1003     0 Feb  6 14:07 sample1.txt
-rw-rw-r--  1 lab1003 lab1003     0 Feb  6 14:07 sample2.txt
-rw-rw-r--  1 lab1003 lab1003     0 Feb  6 14:07 sample3.txt
-rw-rw-r--  1 lab1003 lab1003     0 Feb  6 14:07 sample4.txt
-rw-rw-r--  1 lab1003 lab1003     0 Feb  6 14:07 sample5.txt
-rw-rw-r--  1 lab1003 lab1003   22 Feb  6 14:02 Test1.txt
-r----- 1 lab1003 lab1003  28 Jan 30 15:39 text2.txt
-rw-r----- 1 lab1003 lab1003 12288 Jan 30 16:10 .text2.txt.swp
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$
```

Cd : The cd command in Unix (and Unix-like operating systems) is used to change directories in the file system. It allows you to navigate between different folders within the system.

```
File Edit View Search Terminal Help
lab1003@lab1003-HP-280-G2-MT:~$ mkdir Pratham_63
lab1003@lab1003-HP-280-G2-MT:~$ cd Pratham_63
```

Pwd : The pwd command in Unix (and Unix-like operating systems) stands for Print Working Directory. It is used to display the current working directory, i.e., the full absolute path of the directory you are currently in

```
) ,30(dip),44(video),46(plugdev),100(users),10/( 
pratham10@DESKTOP-K9GVH1:~/Pratham$ pwd
/home/pratham10/Pratham
```

Cat : The cat command in Unix (and Unix-like operating systems) is short for concatenate. It is primarily used to display the contents of a file, but it can also be used to combine multiple files and create new ones.

```
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ vi Hello.sh
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ vi Name.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ cat Name.txt
Name : Pratham
Rollno : 63
Batch : s13
Class : s1
```

Mkdir : The **mkdir** command in Unix (and Unix-like operating systems) is used to create new directories. It's a simple and commonly used command for organizing your file system.

```
File Edit View Search Terminal Help  
lab1003@lab1003-HP-280-G2-MT:~$ mkdir Pratham_63  
lab1003@lab1003-HP-280-G2-MT:~$ cd Pratham_63
```

Rmdir : The **rmdir** command in Unix (and Unix-like operating systems) is used to remove empty directories. It will only delete directories that are empty—if a directory contains files or other directories, you must use other commands like **rm** to remove its contents first.

OR COMMANDS LIKE **rm** TO REMOVE ITS CONTENTS FIRST.

```
ab1003@lab1003-OptiPlex-3070:~$ cd Desktop  
ab1003@lab1003-OptiPlex-3070:~/Desktop$ rmdir Mehta  
ab1003@lab1003-OptiPlex-3070:~/Desktop$ cd..
```

Rm : The **rmdir** command in Unix (and Unix-like operating systems) is used to remove empty directories. It will only delete directories that are empty—if a directory contains files or other directories, you must use other commands like **rm** to remove its contents first

```
lab1003@lab1003-OptiPlex-3070:~$ cd Desktop  
lab1003@lab1003-OptiPlex-3070:~/Desktop$ rm assign7  
lab1003@lab1003-OptiPlex-3070:~/Desktop$ █
```

Cp : The **cp** command in Unix (and Unix-like operating systems) is used to copy files or directories from one location to another

```
lab1003@lab1003-OptiPlex-3070:~/Desktop$ rm assign7  
lab1003@lab1003-OptiPlex-3070:~/Desktop$ cp out.tr Mehta -r  
lab1003@lab1003-OptiPlex-3070:~/Desktop$ █
```

Mv : The **mv** command in Unix (and Unix-like operating systems) is used to move or rename files and directories. It's a versatile command that can either relocate files from one place to another or rename them within the same directory.

```
lab1003@Lab1003-OptiPlex-3070:~/Desktop$ cp out.tr Mehta -r  
lab1003@lab1003-OptiPlex-3070:~/Desktop$ mv file1.odt 123  
lab1003@lab1003-OptiPlex-3070:~/Desktop$ █
```

Chmod : The **chmod** command in Unix (and Unix-like operating systems) is used to change the permissions of files or directories. It controls who can read, write, or execute a file, as well as who has access to a directory.

```
No manual entry for 1
(Alternatively, what manual page do you want from section 1?)
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ chmod 777 Test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ ls -l
total 20
-rw-rw-r-- 1 lab1003 lab1003 49 Jan 30 16:09 Hello1.sh
-rw-rw-r-- 1 lab1003 lab1003 33 Jan 30 16:01 Hello.sh
-rw-rw-r-- 1 lab1003 lab1003 51 Jan 30 15:28 Name.txt
-rw-rw-r-- 1 lab1003 lab1003 0 Feb 6 14:07 sample1.txt
-rw-rw-r-- 1 lab1003 lab1003 0 Feb 6 14:07 sample2.txt
-rw-rw-r-- 1 lab1003 lab1003 0 Feb 6 14:07 sample3.txt
-rw-rw-r-- 1 lab1003 lab1003 0 Feb 6 14:07 sample4.txt
-rw-rw-r-- 1 lab1003 lab1003 0 Feb 6 14:07 sample5.txt
-rwxrwxrwx 1 lab1003 lab1003 22 Feb 6 14:02 Test1.txt
-r----- 1 lab1003 lab1003 28 Jan 30 15:39 text2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$
```

Wc : The **wc** command in Unix (and Unix-like operating systems) stands for word count. It is used to count the number of lines, words, and characters in a file or input stream. It's useful for getting quick statistics about a file's contents

```
lab1003@lab1003-OptiPlex-3070:~/Desktop$ wc salary.txt
9 9 140 salary.txt
```

Piping : Piping in Unix and Linux allows you to send the output of one command directly as the input to another command. This is done using the pipe symbol (|). Piping is a powerful tool for chaining commands together to perform complex operations without the need to create intermediate files.

```
lab1003@lab1003-OptiPlex-3070:~/Desktop$ ls | sort
123
131.nam
131Prac.nam
131Prac.tcl
131Prac.tr
34a.lst
34a.txt
34b.sh
34.lst
Adarsh
Advanced1.pl
Advanced2.pl
assignment7.sh
assignment8
assignment8.sh
Awk
Basics1.pl
Basics2.pl
commands list.odt
evencolumns.txt
fact.sh
first.sh
grep
Mehta
out.nam
output.tr
out.tr
PERL_BASIC1.pl
PracExam
s1105
s12-38
salary.txt
Screenshot from 2024-02-07 11-42-34.png
sid
split_filead
```

Redirection : Redirection in Unix (and Unix-like systems) allows you to control where the output of a command goes, or where input comes from. By default, commands output to the terminal (standard output), but redirection lets you send that output to a file or another command. Similarly, you can also redirect input to a command.

```
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ echo "Saniya Pratham Chamry">>Sample2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ cat Sample2.txt
Saniya Pratham Chamry
```

Grep : The grep command in Unix (and Unix-like systems) is used to search for specific patterns within files or input. It's one of the most commonly used text-processing utilities. The name grep stands for global regular expression print, and it allows you to search using regular expressions (regex) to find matching lines in a file or stream of text.

```
grep: sample1.txt: No such file or directory
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ grep -i "Hello" Sample2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ ls
Hello1.sh Name.txt sample2.txt sample3.txt sample5.txt Test1.txt.gz xaa
Hello.sh sample1.txt Sample2.txt sample4.txt Test1.gz text2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$
```

File : The file command in Unix is used to determine the type of a file based on its content, not just its extension.

```
Test1.txt: cannot open 'Test1.txt' (No such file or directory)
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ file Sample2.txt
Sample2.txt: ASCII text
```

Tr : The tr command in Unix (and Unix-like systems) stands for translate. It's used to replace or remove characters from a stream of text. It can be used for a variety of tasks, such as changing the case of characters, deleting characters, or squeezing multiple consecutive spaces into a single space.

Echo : The echo command in Unix and Linux is a simple but highly useful command for displaying text or the value of a variable to the terminal. It's often used in scripts, shell commands, and pipelines for producing output.

```
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ echo "Saniya Pratham Chamry">Sample2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ cat Sample2.txt
Saniya Pratham Chamry
```

Sort : The sort command in Unix and Linux is used to sort lines of text files or input in a specified order. By default, sort arranges the lines in ascending order, but you can customize its behavior using various options for sorting numerically, by specific columns, or in reverse order.

```
sort: cannot read: Sample.txt: No such file or directory
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ sort Sample2.txt
Saniya Pratham Chamry
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ cat Sample2.txt
```

Head : The head command in Unix and Linux is used to display the first few lines of a file or input. By default, it shows the first 10 lines, but you can adjust the number of lines displayed using options

```
-rw----- 1 lab1003 lab1003 12288 Jan 30 16:10 .text2.txt.swp
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ head -n 1 Test1.txt
Hello TSEC
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ head -n 2 Test1.txt
Hello TSEC
Hello TSEC
```

tail : The tail command in Unix and Linux is used to display the last few lines of a file or output. By default, it shows the last 10 lines of a file, but you can modify the behavior using various options to display more or fewer lines, follow updates to a file in real time, and more.

```
Hello TSEC
Hello TSEC
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ tail -n2 Test1.txt
Hello TSEC
Hello TSEC
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ chmod +r Test1.txt
```

Diff : The diff command in Unix and Linux is used to compare the contents of two files line by line. It outputs the differences between the files, showing which lines need to be changed, added, or deleted in one file to make it identical to the other

```
diff: Try 'diff -h' for more information.
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ diff Text1.txt Sample2.txt
diff: Text1.txt: No such file or directory
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ diff Test1.txt Sample2.txt
1,2c1
< Hello TSEC
< Hello TSEC
...
> Saniya Pratham Chamry
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ 
```

Comm : The **comm** command in Unix compares two sorted files line by line and outputs the differences in three columns:

```
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ comm Test1.txt Sample2.txt
comm: Test1.txt: No such file or directory
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ comm Test1.txt Sample2.txt
comm: Test1.txt: No such file or directory
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ comm Test1.txt Sample2.txt
comm: Test1.txt: No such file or directory
```

Less : The **less** command in Unix is a pager used to view the contents of a file one screen at a time without opening it in an editor. It is useful for reading large files efficiently.

```
-rwxrwxrwx 1 lab1003 lab1003 22 Feb  6 14:02 Test1.txt
-r----- 1 lab1003 lab1003 28 Jan 30 15:39 text2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ less Test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ more Test1.txt
Hello TSEC
Hello TSEC
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ less -1 Test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ 
```

More : The **more** command in Unix is a pager used to view file contents one screen at a time, similar to less, but with fewer features.

```
-rwxrwxrwx 1 lab1003 lab1003 22 Feb  6 14:02 Test1.txt
-r----- 1 lab1003 lab1003 28 Jan 30 15:39 text2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ less Test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ more Test1.txt
Hello TSEC
Hello TSEC
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ less -1 Test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ 
```

Type : The **type** command in Unix is used to determine how a given command is interpreted by the shell. It reveals whether the command is a built-in shell command, an alias, or an external executable.

```
Sample2.txt: ASCII text
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ type Sample2.txt
bash: type: Sample2.txt: not found
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ type cp
cp is /bin/cp
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ split Sample2.txt
```

Split : The **split** command in Unix is used to divide a large file into smaller parts. It is useful for handling large datasets or breaking files for easier processing.

```
cp is /bin/cp
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ split Sample2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ ls
Hello1.sh Hello.sh Name.txt sample1.txt sample2.txt Sample2.txt sample3.txt sample4.txt sample5.txt Test1.txt.gz text2.txt xaa
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ gzip -c Test1 > Test1.gz
```

Find : The find command in Unix is used to search for files and directories based on various criteria like name, size, type, or modification time. It is powerful and flexible for file management tasks

```
grep: EXCLUDING hidden directory status due to previous errors
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ find Test1.txt
Test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ gzip Test1.txt
```

Touch: The touch command is a standard command used in the UNIX/Linux operating system which is used to create, change and modify the timestamps of a file. Basically, there are two different commands to create a file in the Linux system which are as follows:

```
z 4 22
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ touch sample(1,2,3,4,4).txt
bash: syntax error near unexpected token '('
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ touch sample{1,2,3,4,5}.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ ls
Hello1.sh Hello.sh Name.txt sample1.txt sample2.txt sample3.txt sample4.txt sample5.txt Test1.txt text2.txt
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ ls -l
total 20
```

Tar : The tar command in Unix is used to create, extract, and manage archive files, commonly used for backups and file transfers. It supports various compression methods like gzip and bzip2.

```
--> Saniya Pratham Chamry
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ tar x Test1.txt
tar: Refusing to read archive contents from terminal (missing -f option?)
tar: Error is not recoverable: exiting now
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ tar xvf Test1.txt
tar: This does not look like a tar archive
tar: Exiting with failure status due to previous errors
lab1003@lab1003-HP-280-G2-MT:~/Pratham_63$ find Test1.txt
```

Gzip : The gzip command in Unix is used to compress files using the GNU zip (gzip) compression algorithm, reducing file size while preserving original content.

```
grep: Sample1.txt: No such file or directory
lab1003@lab1003-HP-280-G2-NT:~/Pratham_63$ grep -i "Hello" Sample2.txt
lab1003@lab1003-HP-280-G2-NT:~/Pratham_63$ ls
Hello1.sh Name.txt sample2.txt sample3.txt sample5.txt Test1.txt.gz xaa
Hello.sh sample1.txt Sample2.txt sample4.txt Test1.gz text2.txt
lab1003@lab1003-HP-280-G2-NT:~/Pratham_63$
```