

AIM:1. . Explore the internal commands of linux like

1.ls

ls: It is used to list information about files and directories within the file system.

Example:

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)

\$ ls

Autodesk/ Bing.url desktop.ini Links/

ls -a: In Linux, hidden files start with . (dot) symbol and they are not visible in the regular directory. The (ls -a) command will enlist the whole list of the current directory including the hidden files.

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)

\$ ls -a

```
./                  CSS-Riya/          JQuery/
../                 DOM/              me
'002 Download the Starting Files.html' DOM.zip            my-express-js/
'036 CSS - Bacon Fansite.zip'      'Drum Kit Starting Files'/  Myntra-Clone/
Calculator/                        'Drum Kit Starting Files.zip' Myntra-Clone.zip
Calculator.zip                     gitpic.jpeg             NODEJS/
css/                              hobbies.html            profilepicriya.jpeg
'CSS - Bacon Fansite'/             home1.html              Rubix-23-36-Code-Raiders/
'CSS - Chocolate-Fansite'/         'HTML-PERSONAL WEBSITE'/  Survey/
```

ls -l : It will show the list in a long list format.

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)

\$ ls -l

total 12

```
drwxr-xr-x 1 ROHINI 197609  0 Dec 18 21:54 CSS/
drwxr-xr-x 1 ROHINI 197609  0 Dec 19 00:08 images/
-rw-r--r-- 1 ROHINI 197609 751 Dec 28 21:08 index.html
-rw-r--r-- 1 ROHINI 197609 1311 Dec 28 21:07 index.js
```

ls -lh: This command will show you the file sizes in human readable format. Size of the file is very difficult to read when displayed in terms of byte. The (ls -lh)command will give you the data in terms of Mb, Gb, Tb, etc.

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)

\$ ls -lh

total 12K

```
drwxr-xr-x 1 ROHINI 197609  0 Dec 18 21:54 CSS/
drwxr-xr-x 1 ROHINI 197609  0 Dec 19 00:08 images/
```

```
-rw-r--r-- 1 ROHINI 197609 751 Dec 28 21:08 index.html
```

```
-rw-r--r-- 1 ROHINI 197609 1.3K Dec 28 21:07 index.js
```

ls -lhS: If you want to display your files in descending order (highest at the top) according to their size, then you can use (ls -lhS) command.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)
```

```
$ ls -lhS
```

```
total 12K
```

```
-rw-r--r-- 1 ROHINI 197609 1.3K Dec 28 21:07 index.js
```

```
-rw-r--r-- 1 ROHINI 197609 751 Dec 28 21:08 index.html
```

```
drwxr-xr-x 1 ROHINI 197609 0 Dec 18 21:54 CSS/
```

```
drwxr-xr-x 1 ROHINI 197609 0 Dec 19 00:08 images/
```

ls -d*/ :It is used to display only subdirectories.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)
```

```
$ ls -d */
```

```
CSS// images//
```

ls -g or ls -lg : With this you can exclude column of group information and owner.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)
```

```
$ ls -g
```

```
total 12
```

```
drwxr-xr-x 1 197609 0 Dec 18 21:54 CSS/
```

```
drwxr-xr-x 1 197609 0 Dec 19 00:08 images/
```

```
-rw-r--r-- 1 197609 751 Dec 28 21:08 index.html
```

```
-rw-r--r-- 1 197609 1311 Dec 28 21:07 index.js
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)
```

```
$ ls -lg
```

```
total 12
```

```
drwxr-xr-x 1 197609 0 Dec 18 21:54 CSS/
```

```
drwxr-xr-x 1 197609 0 Dec 19 00:08 images/
```

```
-rw-r--r-- 1 197609 751 Dec 28 21:08 index.html
```

```
-rw-r--r-- 1 197609 1311 Dec 28 21:07 index.js
```

ls -n :It is used to print group ID and owner ID instead of their names.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)
```

```
$ ls -n
```

```
total 12
```

```
drwxr-xr-x 1 197609 197609 0 Dec 18 21:54 CSS/
```

```
drwxr-xr-x 1 197609 197609 0 Dec 19 00:08 images/
```

```
-rw-r--r-- 1 197609 197609 751 Dec 28 21:08 index.html
```

```
-rw-r--r-- 1 197609 197609 1311 Dec 28 21:07 index.js
```

ls -li: This command prints the index number if file is in the first column.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)
```

```
$ ls -li
```

```
total 12
```

```
21110623253461002 drwxr-xr-x 1 ROHINI 197609  0 Dec 18 21:54 CSS/  
47569271064172626 drwxr-xr-x 1 ROHINI 197609  0 Dec 19 00:08 images/  
34339947158754179 -rw-r--r-- 1 ROHINI 197609 751 Dec 28 21:08 index.html  
16888498603012016 -rw-r--r-- 1 ROHINI 197609 1311 Dec 28 21:07 index.js
```

ls -r: It print the list in reverse order.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)
```

```
$ ls -r
```

```
index.js index.html images/ CSS/
```

ls -R: It gives file of subdirectory also.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)
```

```
$ ls -R
```

```
..
```

```
CSS/ images/ index.html index.js
```

```
./CSS:
```

```
cssD1.css CSSQ.css
```

```
./images:
```

```
347920ae-647c-4a00-a8f2-f742ed03932f.jpg image2.png image4.png image6.png  
image1.png image3.png image5.png kitten.png
```

2)MKdir: The command mkdir stands for “make directory”. It creates each directory specified on the command line in the order given.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
```

```
$ mkdir hello.c
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
```

```
$ ls
```

```
Autodesk/ Bing.url desktop.ini hello.c/ Links/
```

-version: displays version number and some information about license.

Syntax:

```
mkdir -version
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)
```

\$ mkdir --version

mkdir (GNU coreutils) 8.32

Copyright (C) 2020 Free Software Foundation, Inc.

License GPLv3+: GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie.

-help: displays help related information.

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)

\$ mkdir --help

Usage: mkdir [OPTION]... DIRECTORY...

Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.

- m, --mode=MODE set file mode (as in chmod), not a=rwx - umask
- p, --parents no error if existing, make parent directories as needed
- v, --verbose print a message for each created directory
- Z set SELinux security context of each created directory to the default type
- context[=CTX] like -Z, or if CTX is specified then set the SELinux or SMACK security context to CTX
- help display this help and exit
- version output version information and exit

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>

Full documentation <<https://www.gnu.org/software/coreutils/mkdir>>

or available locally via: info '(coreutils) mkdir invocation'

-v or --verbose: It displays a message for every directory created.

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)

\$ mkdir -v ria

mkdir: created directory 'ria'

-p: A flag which enables the command to create parent directories as necessary. If the directories exist, no error is specified.

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)

\$ mkdir -p -v hi/hello

mkdir: created directory 'hi'

mkdir: created directory 'hi/hello'

-m: This option is used to set the file modes, i.e. permissions, etc. for the created directories. The syntax of the mode is the same as the chmod command.

syntax:`mkdir -m a=rwx [directories]`

The above syntax specifies that the directories created give access to all the users to read from, write to and execute the contents of the created directories. You can use 'a=r' to only allow all the users to read from the directories and so on.

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/DOM (main)

\$ `mkdir -m a=rwx gfg`

3)chdir:In Linux, the chdir command is used to **change the current working directory**. This command is similar to the cd command in Unix.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
```

```
$ cd hello.c
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/hello.c (main)
```

```
$ ^C
```

Cd / :gives root directory

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
```

```
$ cd /
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 /
```

```
$ ^C
```

Cd ~ :gives home/main directory

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~ (main)
```

```
$ cd /
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 /
```

```
$ cd ..
```

Cd .. :one level up

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
```

```
$ cd ..
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~ (main)
```

```
$
```

Cd - :to change previous directory

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~ (main)
$ cd -
/c/Users/ROHINI/webdevelopmentRiya

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$
```

4)rmkdir:The rmkdir command is used to remove empty directories from our Linux Operating System. Every directory that needs to be removed should not contain any files or subdirectories. Otherwise, the rmkdir command cannot remove the directory from our file system.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
$ rmdir hello.c

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
$ ls
Autodesk/ Bing.url desktop.ini Links/
```

Example 1: This will first remove the child directory and then remove the parent directory.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya/mydir (main)
$ cd -
/c/Users/ROHINI/webdevelopmentRiya

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ rmdir -p mydir/mydir1
```

Example 2: Remove the directories mydir1, mydir2, and mydir3, if they are empty. If any of these directories are not empty, then an error message will be printed for that directory, and the other directories will be removed.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ rmdir mydir1 mydir2

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ ls
'002 Download the Starting Files.html'  DOM.zip          my-express-js/
'036 CSS - Bacon Fansite.zip'          'Drum Kit Starting Files/'  Myntra-Clone/
Calculator/                             'Drum Kit Starting Files.zip' Myntra-Clone.zip
Calculator.zip                          gitpic.jpeg         NODEJS/
css/                                    hobbies.html        profilepicriya.jpeg
'CSS - Bacon Fansite/'                  home1.html          Rubix-23-36-Code-Raiders/
'CSS - Chocolate-Fansite/'              'HTML-PERSONAL WEBSITE'/    Survey/
CSS-Riya/                               JQuery/
DOM/                                     me
```

Example 3: Remove the directory mydir/mydir1 if it is empty. Then, remove directory mydir, if it is empty after mydir/mydir1 was removed.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ rmdir mydir/mydir1 mydir
```

5)cat: The cat command is termed “concatenate”. It performs three main roles related to manipulation of text files: creating them, displaying them & combining them.

1)cat filename: To view file and its content

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat home1.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CONTACTRiyaJaiswal</title>
  <link rel="stylesheet" href="/css/styles.css">

</head>
```

```
<body>
  <h3>CONTACT DETAILS</h3>
  <p>PH NO:9*****<p>
  <p>EMAIL ID:miss.riyajaiswal251003@gmail.com</p>
  <p>ADDRESS:Mumbai<p>
  <hr>
  <h4>Submit your Infos to get Response:</h4>
  .....
</html>
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
```

2)cat file1 file2 : To view multiple files

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat home1.html hobbies.html
<!DOCTYPE html>
<html lang="en">
<head> .....
  </form></body>
</html><!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  .....
  .....
</body>
</html>
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$
```

3)cat -n filename:To view contents of a file preceding with line numbers.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat -n hobbies.html
  1 <!DOCTYPE html>
  .....
 19 </body>
 20 </html>
```



```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$
```

4)**cat > newfile**:To create a new file and write in it through cmd line,ctrl+d to come out.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat > meriya
this is mysterious girl!!
linux cmds are fun untill u have to do assngmnt

ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$
```

5)**\$cat [filename-whose-contents-is-to-be-copied] > [destination-filename]** : Copy the contents of one file to another file. Command.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat m1.txt >m2.txt
```

6)**\$cat -s geeks.txt** : Cat command can suppress repeated empty lines in output

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat -s m1.txt
```

7)**\$cat file1 >> file2** : Cat command can append the contents of one file to the end of another file.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat m1.txt >> m2.txt
ROHINI@DESKTOP-AAJD063 M
```

8)**\$tac filename** :Cat command can display content in reverse order using tac command.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ tac m1.txt
```

9) **cat -E "filename"** : Cat command can highlight the end of line.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat -E "me"
$
cd -$
$
$
$
ROH
```

10) **\$cat -A "filename"**: If you want to use the -v, -E and -T option together, then instead of writing -vET in the command, you can just use the -A command line option.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat -A "meriya"
this is mysterious girl!!$
linux cmds are fun untill u have to do assngmnt$
```

11) **Cat command to write in an already existing file.**

Cat >> filename

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat >> meriya
I am tired
thats all
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$ cat meriya
this is mysterious girl!!
linux cmds are fun untill u have to do assngmnt
I am tired
thats all
ROHINI@DESKTOP-AAJD063 MINGW64 ~/webdevelopmentRiya (main)
$
```

6) **rm:rm** command in UNIX stands for remove and by default is used for removing files. It is simple but a powerful command especially when used with options such as -rf which allow it to delete non-empty directories forcefully.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
$ mkdir A
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
$ touch a.txt
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
$ ls
A/ a.txt Autodesk/ Bing.url desktop.ini Links/
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
$ rm a.txt
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
$ rm A
rm: cannot remove 'A': Is a directory
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
$ ls
A/ Autodesk/ Bing.url desktop.ini Links/
```

7)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.It has two distinct functions:

- (i) It renames a file or folder.
- (ii) It moves a group of files to a different directory.

```
rename:
$ ls
a.txt b.txt c.txt

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Trial (main)
$ mv a.txt hr.txt

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Trial (main)
$ ls
b.txt c.txt hr.txt

_____

move:
$ ls
a.txt b.txt c.txt

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Trial (main)
$ mv a.txt hr.txt

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Trial (main)
$ ls
b.txt c.txt hr.txt
```

8)cp:cp cmd copy one file to another.

Syntax:cp source destination.

Example

```
ROHINI@DESKTOP-AAJD063 MINGW64
~/webdevelopmentRiya (main)
$ cp meriya m1.txt

ROHINI@DESKTOP-AAJD063 MINGW64
~/webdevelopmentRiya (main)
$ cat << m1.tet
>
bash: warning: here-document at line 74 delimited by
end-of-file (wanted `m1.tet')

ROHINI@DESKTOP-AAJD063 MINGW64
~/webdevelopmentRiya (main)
$ cat m1.txt
```

```
this is mysterious girl!!
linux cmds are fun untill u have to do assngmnt
I am tired
thats all
ROHINI@DESKTOP-AAJD063 MINGW64
~/webdevelopmentRiya (main)
$ cat meriya
this is mysterious girl!!
linux cmds are fun untill u have to do assngmnt
I am tired
thats all
ROHINI@DESKTOP-AAJD063 MINGW64
~/webdevelopmentRiya (main)
```

9)head:The head command, as the name implies, print the top N number of data of the given input. By default, it prints the first 10 lines of the specified files. If more than one file name is provided then data from each file is preceded by its file name.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
$ cat > b.txt //editing file
riya
jaiswal
krishna
xyz
aarti
myself
os
linux
unix

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
$ head b.txt //default 10 lines it is printing
riya
jaiswal
krishna
xyz
aarti
myself
os
linux
unix
```

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
$ head -n 6 b.txt //6—>6 lines are printed
riya
jaiswal
krishna
xyz
aarti
myself
```

10)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.

```
ROHINI@DESKTOP-AAJD063 MINGW64
~/Favorites/Links/Trial (main)
$ tail b.txt
riya
jaiswal
krishna
xyz
aarti
myself
Os
linux
unix

ROHINI@DESKTOP-AAJD063 MINGW64
~/Favorites/Links/Trial (main)
$ tail -n 3 b.txt
os
linux
unix
```

11)sort: Sort is a standard command-line program that prints the lines of its input or concatenation of all files listed in its argument list in sorted order. The sort command is a command-line utility for sorting lines of text files. It supports sorting alphabetically, in reverse order, by number, by month, and can also remove duplicates.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
$ sort b.txt
aarti
jaiswal
krishna
linux
myself
os
riya
unix
xyz
```

12)wc :wc command in Linux with examples. wc stands for word count. As the name implies, it is mainly used for counting purpose. It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments. By default it displays four-columnar output.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
$ wc b.txt
 9  9 52 b.txt
```

13)chown:The basic chown command syntax consists of a few segments. The help file shows the following format:

```
chown [OPTIONS]
USER[:GROUP] FILE(s)
```

- [OPTIONS] – the command can be used with or without additional options.
- [USER] – the username or the numeric user ID of the new owner of a file.
- [:] – use the colon when changing a group of a file.
- [GROUP] – changing the group ownership of a file is optional.
- FILE – the target file.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
$ chown --version
chown (GNU coreutils) 8.32
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie and Jim Meyering.
```

14)chmod:In Unix-like operating systems, the chmod command is used to change the access mode of a file.

syntax:chmod [reference][operator][mode] file...

The references are used to distinguish the users to whom the permissions apply i.e. they are list of letters that specifies whom to give permissions.

R-read permission

W-write permission

X-execute permission

```
ROHINI@DESKTOP-AAJD063 MINGW64  
~/webdevelopmentRiya (main)  
$ chmod u=rwx meriya
```

```
ROHINI@DESKTOP-AAJD063 MING
```

15)chgrp:chgrp command in Linux is used to change the group ownership of a file or directory. All files in Linux belong to an owner and a group. You can set the owner by using “chown” command, and the group by the “chgrp”

command.Syntax:

```
chgrp [OPTION]... GROUP FILE...
```

```
chgrp [OPTION]... -reference=RFILE FILE...
```

```
chgrp -hR www-data /var/www
```

16)umask: The umask command in Linux is used to set default permissions for files or directories the user creates. The umask command specifies the permissions that the user does not want to be given out to the newly created file or directory.

umask works by doing a Bitwise AND with the bitwise complement(where the bits are inverted, i.e. 1 becomes 0 and 0 becomes 1) of the umask. The bits which are set in the umask value, refer to the permissions, which are not assigned by default, as these values are subtracted from the maximum permission for files/directories.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)  
$ umask  
0022
```

17)ps:The ps command, short for Process Status, is a command line utility that is used to display or view information related to the processes running in a Linux system. As we

all know, Linux is a multitasking and multiprocessing system. Therefore, multiple processes can run concurrently without affecting each other.

```
ROHINI@DESKTOP-AAJD063 MINGW64
ROHINI@DESKTOP-AAJD063 MINGW64
~/Favorites/Links/Trial (main)
$ PS
  PID  PPID  PGID  WINPID  TTY      UID   STIME
COMMAND
  1340    1  1340   12148  cons0    197609 19:27:56
/usr/bin/bash
  2123  1340   2123    7996  cons0    197609 22:41:52
/usr/bin/PS
```

18)pipe:A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux and other Unix-like operating systems to send the output of one command/program/process to another command/program/process for further processing. The Unix/Linux systems allow stdout of a command to be connected to stdin of another command. You can make it do so by using the pipe character '|'. Pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on. It can also be visualized as a temporary connection between two or more commands/ programs/ processes.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
$ sort b.txt | uniq
aarti
jaiswal
krishna
linux
myself
os
riya
unix
xyz
```

19)redirection:Redirection is a feature in Linux such that when executing a command, you can change the standard input/output devices. The basic workflow of any Linux command is that it takes an input and give an output.

- The standard input (stdin) device is the keyboard.
- The standard output (stdout) device is the screen.

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
$ date > c.txt

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
$ cat c.txt
Fri, Jan 27, 2023 11:16:57 PM
```

20.

a) Display top 10 processes in descending order:

```
tsec11@ubuntu:~$ ps -e | head -n 10
```

PID	TTY	TIME	CMD
1	?	00:00:03	systemd
2	?	00:00:00	kthreadd
3	?	00:00:00	rcu_gp
4	?	00:00:00	rcu_par_gp
5	?	00:00:00	kworker/0:0-eve
6	?	00:00:00	kworker/0:0H-kb
7	?	00:00:00	kworker/u256:0-
8	?	00:00:00	mm_percpu_wq
9	?	00:00:00	ksoftirqd/0
10	?	00:00:00	rcu_sched

b) Display the processor with highest memo usage

```
tsec11@ubuntu:~$ ps -eo pid,ppid,%mem,%cpu --sort=-%mem | head -n 10
```

PID	PPID	%MEM	%CPU
2128	1980	7.9	1.9
1181	1112	6.0	0.5
2479	1951	5.9	0.6
2645	1	5.2	1.3
2555	1	2.8	0.1
1971	1969	2.8	0.5
2347	1951	2.7	0.1
2395	2347	2.5	0.0
1496	1112	2.4	0.0

c) Display current user logged in and logname

```
tsec11@ubuntu:~$ echo "your Logname:$(echo $LOGNAME)"
your Logname:tsec11
```

d)

Display current shell, home directory, operating system type, current path setting, current working directory,

```
tsec11@ubuntu:~$ echo "Current shell:$(echo $SHELL)"
Current shell:(/bin/bash)
```

```
tsec11@ubuntu:~$ echo "your username:$(echo $USER)"
your username:(tsec11)
```

```
//cd pwd (gives present working directory)  
//cd ~(gives home / main directory)
```

e)Display OS version, release number, kernel version.

```
tsec11@ubuntu:~$ uname -a #displays information about the current system  
hardware platform ,name of OS and its version and so on  
Linux ubuntu 5.4.0-137-generic #154~18.04.1-Ubuntu SMP Tue Jan 10 16:58:20  
UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```