**PRACTICAL-3**

Q. Write C programs to simulate the following unix file commands:

**A.FILE**

**1.Ls**

The ls command is used to list all files in the current directory.

The -l argument lists the files in ‘long format’, which contains lots of useful information, eg. The exact size of the file, who owns the file and who has the right to look at it.

The -a argument lists all files, including the ones whose filenames begin in a dot, which we do not always see.

**2.More**

The more command shows the firstpart of a file, just as much as will firt on one screen.Just hit the space bar to see more or q to quit.

Syntax- more filename

**3.Emacs**

Emacs is an editor that lets us create and edit a file.

**4.Mv**

The mv command is used to move a file(ie gives it a different name, or moves it into a different directory).

Syntax- mv filename1 filename2

**5.Cp**

The cp command copies a file.

Syntax- cp filename1 fielname2

**6.Rm**

The rm command is used to remove a file.

Syntax- rm filename

**7.Diff**

It compares files and shows where they differ.

**8.Wc**

It tells us how many lines, words and characters there are in a file.

**9.Chmod**

It lets us change the read, write and execute permissions on your files. The default is that we can only look at them and change them, but we may sometimes want to change these permissions.

Code:

#include<stdio.h>

#include<dirent.h>

#include<string.h>

#include<sys/stat.h>

#include<sys/types.h>

#include<unistd.h>

#include<stdlib.h>

int main(int argc, char\* argv[])

{

if(argc!=2)

{

printf("Enter file name.\n");

return 1;

}

DIR \*p;

struct dirent \*d;

p=opendir(argv[1]);

if(p==NULL)

{

printf("Cannot find directory\n");

return 1;

}

printf("Viewing results for ls command:\n");

while(d=readdir(p))

{

if(d->d\_name[0] != '.')

printf("%s\t",d->d\_name);

}

closedir(p);

p= opendir(argv[1]);

printf("\nViewing results for ls -a command:\n");

while(d=readdir(p))

printf("%s\t", d->d\_name);

closedir(p);

char buffer[512];

struct stat buf;

p = opendir(argv[1]);

printf("\nViewing results for ls -l command:\n");

while((d = readdir(p)) != NULL)

{

sprintf(buffer, "%s/%s", argv[1], d->d\_name);

stat(buffer, &buf);

printf("%ld",buf.st\_size);

printf(" %s\t\n", d->d\_name);

}

closedir(p);

printf("Copying a file hw\n");

system("cp hw.txt aa.txt");

printf("More on a file:\n");

system("more - hw.txt");

printf("Moving a file hw.txt to ab.\n");

system("mv hw.txt ab.txt");

printf("Removing file ab.\n");

system("rm ab.txt");

printf("Comparing hw.txt and aa.txt.\n");

system("diff hw.txt aa.txt");

printf("Making b executable.\n");

system("chmod +x b");

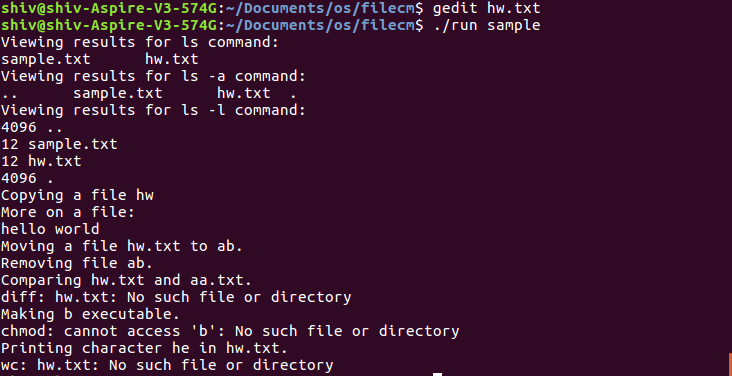
printf("Printing character he in hw.txt.\n");

system("wc -m hw.txt");

return 0;

}

Output:



**B.DIRECTORY**

**1.Mkdir**

The mkdir command is used to make a new directory.

Syntax- mkdir dirname

where dirname is the name of the directory to be created.

**2.Cd**

The cd command is use dto change directory. We basically go to another directory.

Syntax- cd dirname

where dirname is the name of the directory we want to go to.

**3.Pwd**

The pwd command gives us the name of the current directory. That is, it tells us where we currently are.

Syntax- pwd

Code:

#include<stdio.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<unistd.h>

int main(int argc, char\* argv[])

{

if(argc!=2)

{

printf("Enter file name.\n");

return 1;

}

int val;

char \*dname = argv[1];

val = mkdir(dname,0700);

if(!val)

printf("Directory created succesfully!\n");

else

{

printf("Unable to create directory.\n");

return 1;

}

int val2;

char path[100];

snprintf(path,99,"/home/ananya/Practical/%s", dname);

val2= chdir(path);

if(val2==0)

printf("Directory changed succesfully!\n");

else

return 1;

char cwd[200];

if(getcwd(cwd, sizeof(cwd))!= NULL)

printf("Current working directory is: %s\n", cwd);

else

return 1;

return 0;

}

Output:

