**40. Illustrate the various File Access Permission and different types users in Linux.**

**Aim:** To write a c program about various File Access Permission and different types users in Linux.

# **ALGORITHM:**

Create a file or identify an existing file to demonstrate permissionsand users.

View the file's permissions using the ls -l command. The outputwill look something like this:

.txt

The first character (-) represents the file type (a dash indicates aregular file).

The next three characters (rw-) represent the permissions for thefile's owner (Read and Write, no Execute).

The next three characters (r--) represent the permissions for thefile's group (Read, no Write or Execute).

The last three characters (r--) represent the permissions for others(Read, no Write or Execute).

The number 1 represents the number of hard links to the file.

owner is the username of the file's owner.

group is the name of the file's group.

1234 is the file's size in bytes.

Oct 19 10:30 is the last modification timestamp.

file.txt is the file name.

Use the chmod command to change the file's permissions. Forexample, to give the group write permission, use chmod g+w file.txt.

Re-run ls -l to confirm the updated permissions.

You can also change the file's owner and group using the chownand chgrp commands, respectively.

To create and manage user accounts, you can use the useradd andpasswd commands.

**Program:** #include <stdio.h> #include<stdlib.h>

#include <string.h> #include <unistd.h>

int main(int argc, char \*\*argv) { int result;

char \*filename = (char \*)malloc(512); if (argc < 2) {

strcpy(filename, "/usr/bin/adb");

} else {

strcpy(filename, argv[1]);

}

result = access (filename, R\_OK); if ( result == 0 ) {

printf("%s is readable\n",filename);

} else {

printf("%s is not readable\n",filename);

}

result = access (filename, W\_OK); if ( result == 0 ) {

printf("%s is Writeable\n",filename);

} else {

printf("%s is not Writeable\n",filename);

}

result = access (filename, X\_OK);

if ( result == 0 ) {

printf("%s is executable\n",filename);

} else {

printf("%s is not executable\n",filename);

}

free(filename); return 0;

}

**Output:**

