

NAME: G. GANESH

REG NO: 192373008

EXERICSE-40

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

AIM:

To illustrate file access permissions and different types of users in Linux using a C program that demonstrates user, group, and others' permissions (read, write, execute) on a file.

Algorithm:

1. Setup File Details:

- Specify the file name to analyze.
- Retrieve the file's permission details using system calls.

2. Retrieve File Metadata:

- Use the stat() system call to obtain file information.
- Extract details about ownership, file type, and permissions.

3. Analyse Permissions:

- Check read, write, and execute permissions for **user**, **group**, and **others**.
- Display permissions in a human-readable format (e.g., rwx).

4. Display Results:

- Print the file permissions, file type, and associated user and group IDs.

Procedure:

1. Include necessary headers (stdio.h, sys/stat.h, unistd.h).
2. Define a function to convert permission bits into a readable format.
3. Use the stat() system call to retrieve metadata of a file.
4. Parse and display file type and permission details.
5. Compile and run the program with a valid file name as input.

Code:

```
#include <stdio.h>
```

```
#include <sys/stat.h>
```

```
#include <stdlib.h>
```

```
void print_permissions(mode_t mode) {
```

```

printf((mode & S_IRUSR) ? "r" : "-");
printf((mode & S_IWUSR) ? "w" : "-");
printf((mode & S_IXUSR) ? "x" : "-");
printf((mode & S_IRGRP) ? "r" : "-");
printf((mode & S_IWGRP) ? "w" : "-");
printf((mode & S_IXGRP) ? "x" : "-");
printf((mode & S_IROTH) ? "r" : "-");
printf((mode & S_IWOTH) ? "w" : "-");
printf((mode & S_IXOTH) ? "x" : "-");
printf("\n");
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: %s <file_name>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    struct stat fileStat;
    if (stat(argv[1], &fileStat) < 0) {
        perror("Error retrieving file information");
        exit(EXIT_FAILURE);
    }

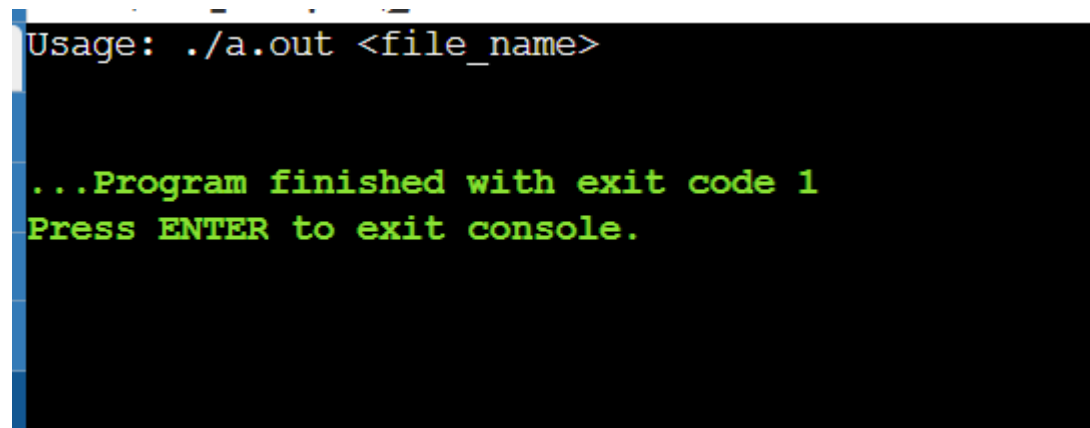
    printf("File: %s\n", argv[1]);
    printf("File Type: ");
    if (S_ISREG(fileStat.st_mode)) printf("Regular File\n");
    else if (S_ISDIR(fileStat.st_mode)) printf("Directory\n");
    else if (S_ISLNK(fileStat.st_mode)) printf("Symbolic Link\n");
    else printf("Other\n");
    printf("Owner UID: %d\n", fileStat.st_uid);
    printf("Group GID: %d\n", fileStat.st_gid);
    printf("Permissions: ");

```

```
    print_permissions(fileStat.st_mode);  
    return 0;  
}
```

Result:

The program demonstrates file permissions and different types of users (owner, group, others) in Linux, illustrating their access levels in a human-readable format.

Output:A terminal window with a black background and green text. The first line shows the usage: 'Usage: ./a.out <file_name>'. The second line shows the program's exit message: '...Program finished with exit code 1'. The third line shows the instruction: 'Press ENTER to exit console.'

```
Usage: ./a.out <file_name>  
  
...Program finished with exit code 1  
Press ENTER to exit console.
```