**3.4** Implement a program to print the various types of file in Linux.

**Objectives:**

1. To learn about File system Internals.

**Theory:**

stat(pathname, &sb);

if ((sb.st\_mode & S\_IFMT) == S\_IFREG) {

/\* Handle regular file \*/

}

Because tests of the above form are common, additional macros are defined by POSIX to allow the test of the file type in st\_mode to be written more concisely:

S\_ISREG(m) is it a regular file?

S\_ISDIR(m) directory?\newline

S\_ISCHR(m) character device?

S\_ISBLK(m) block device?

S\_ISFIFO(m) FIFO (named pipe)?

S\_ISLNK(m) symbolic link? (Not in POSIX.1-1996.)

S\_ISSOCK(m) socket? (Not in POSIX.1-1996.)

**Data Dictionary:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr Number | Variable/Function | Datatype | Use |
|  |  |  |  |
| 1 | file | int | Open a file and store file descriptor. |
|  |  |  |  |
| 2 | fileStat | struct stat | Store information about files. |
|  |  |  |  |

**Program:**

#include <unistd.h>

#include <fcntl.h>

#include <stdio.h>

#include <sys/stat.h>

#include <sys/types.h>

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

{

if(argc != 2)

return 1;

int file;

if((file=open(argv[1],O\_RDONLY)) < -1)

return 1;

struct stat fileStat;

if(fstat(file,&fileStat) < 0)

return 1;

if(S\_ISBLK(fileStat.st\_mode))

{

printf("Block special file\n");

}

if(S\_ISCHR(fileStat.st\_mode))

{

printf("Character special file\n");

}

if(S\_ISDIR(fileStat.st\_mode))

{

printf("Is a directory\n");

}

if(S\_ISFIFO(fileStat.st\_mode))

{

printf("Pipes and FIFO\n");

}

if(S\_ISLNK(fileStat.st\_mode))

{

printf("Is a symbolic link\n");

}

if(S\_ISREG(fileStat.st\_mode))

{

printf("A Regular File\n");

}

if(S\_ISSOCK(fileStat.st\_mode))

{

printf("Integrated Sockets\n");

}

return 0;

}

**Conclusion:**

Various types of files like block file, character file, regular file, FIFO and pipe can be studied and checked with the status of different flags.

**References:**

https://www.cyberciti.biz/faq/linuxunix-determine-file-type/