**Chapter 3 : File system Internals**

* 1. **- Write the program to show file statistics using the stat system call. Take the filename / directory name from user including path.**

**Objectives :**

* + 1. To learn about File system Internals.

**Theory :**

Name:

stat, fstat, lstat - get file status

Syntax:

#include <sys/types.h> #include <sys/stat.h> #include <unistd.h>

int stat(const char \**path*, struct stat \**buf*); int fstat(int *fd*, struct stat \**buf*);

int lstat(const char \**path*, struct stat \**buf*);

Description:

These functions return information about a file. No permissions are required on the file itself, but-in the case of stat() and lstat() - execute (search) permission is required on all of the directories in *path* that lead to the file.

stat() stats the file pointed to by *path* and fills in *buf*.

lstat() is identical to stat(), except that if *path* is a symbolic link, then the link itself is stat-ed, not the file that it refers to.

fstat() is identical to stat(), except that the file to be stat-ed is specified by the file descriptor *fd*.

All of these system calls return a *stat* structure, which contains the following fields : struct stat {

dev\_t st\_dev; /\* ID of device containing file \*/ ino\_t st\_ino; /\* inode number \*/

mode\_t st\_mode; /\* protection \*/ nlink\_t st\_nlink; /\* number of hard links \*/

|  |  |  |
| --- | --- | --- |
| uid\_t | st\_uid;  st\_gid; | /\* user ID of owner \*/ gid\_t  /\* group ID of owner \*/ |
| dev\_t | st\_rdev; | /\* device ID (if special file) \*/ |
| off\_t | st\_size; | /\* total size, in bytes \*/ |

blksize\_t st\_blksize; /\* blocksize for file system I/O \*/ blkcnt\_t st\_blocks; /\* number of 512B blocks allocated \*/ time\_t st\_atime; /\* time of last access \*/

time\_t st\_mtime; /\* time of last modification \*/ time\_t st\_ctime; /\* time of last status change \*/

};

**Data Dictionary :**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr Number | Variable/Function | Datatype | Use |
| 1 | fileStat | struct stat | Store information about files. |

**Flowchart :**

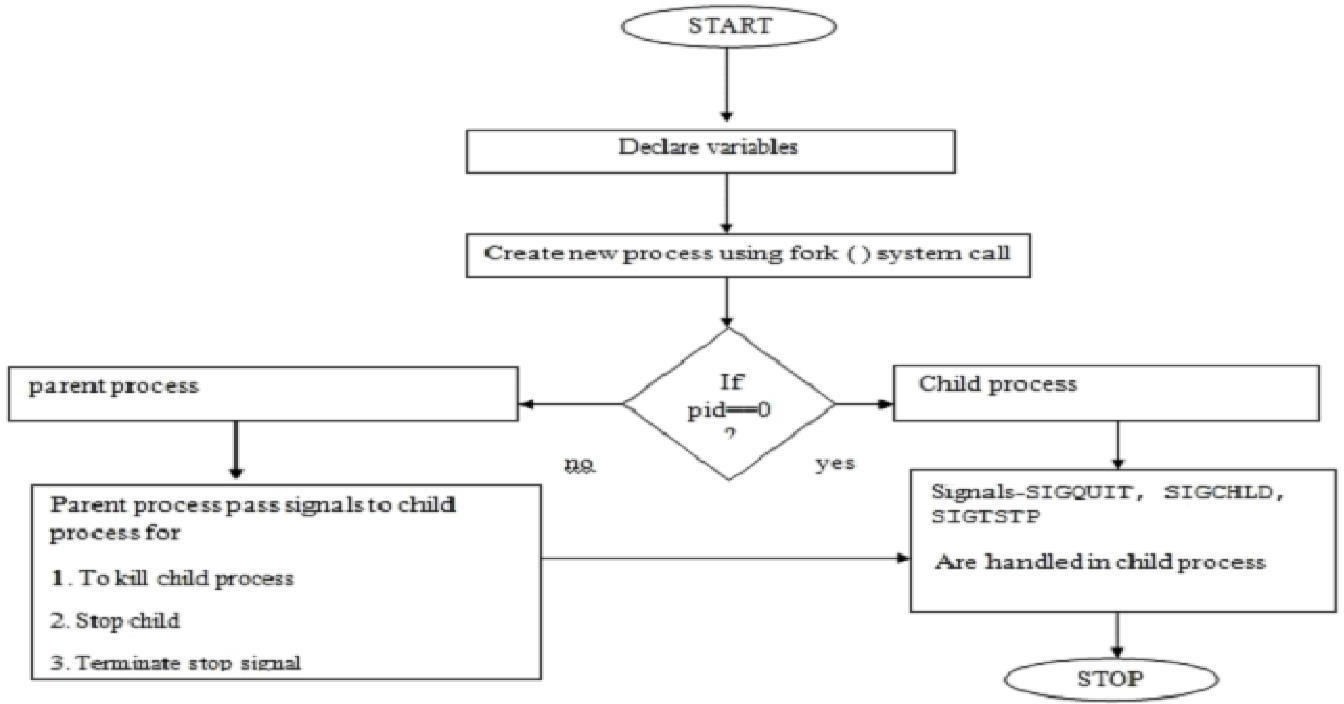
****

Fig: 3.1 Flowchart

**Program :**

**#include <stdio.h>**

#include <stdlib.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

int main() {

char path[100];

printf("Enter the file or directory path: ");

fgets(path, sizeof(path), stdin);

// Remove the newline character from the input

if (path[strlen(path) - 1] == '\n') {

path[strlen(path) - 1] = '\0';

}

struct stat fileStat;

// Use the stat system call to retrieve file statistics

if (stat(path, &fileStat) == -1) {

perror("Error: Unable to retrieve file statistics");

exit(EXIT\_FAILURE);

}

printf("File Statistics:\n");

printf("Path: %s\n", path);

printf("Size: %ld bytes\n", fileStat.st\_size);

printf("Owner ID: %u\n", fileStat.st\_uid);

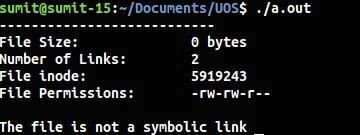
printf("Group ID: %u\n", fileStat.st\_gid);

printf("File Permissions: %o\n", fileStat.st\_mode);

return 0;

}

**Output :**

****

**Conclusion :**

* Stats of file like file size,links, permissions, inode number and type of link can be retrieved using stat() and stored in a structure.

**References :**

[1]https:/[/www.lix.polyt](http://www.lix.polytechnique.fr/~liberti/public/computing/prog/c/C/FUNCTIONS/stat.h)e[chnique.fr/~liberti/public/computing/prog/c/C/FUNCTIONS/stat.h](http://www.lix.polytechnique.fr/~liberti/public/computing/prog/c/C/FUNCTIONS/stat.h)