File system Internals

3B: File system Internals: stat, fstat, ustat, link/unlink,dup

Subject:- Unix Operating
System Lab Class :- TYIT
Name : Aditi Sudhir Ghate
PRN : 2020BTEIT00044

3.b Write the program to show file statistics using the fstat system call. Take the file name / directory name from user including path. Print only inode no, UID, GID, FAP and File type only.

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; /* user ID of owner */

gid_t st_gid; /* 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 Varia [		
1	s	char[]	Get file name.
2	fp	FILE*	Pointer to file.
3	fn	int	File descriptor number.
4	sta		at Store information about

Program:

```
#include<stdio.h>

#include<stdlib.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <unistd.h>

int main()

{

 char s[100];

 gets (s);

 //printf("%s",s);

FILE *fp;

//link/unlink,dup;

if((fp=fopen(s,"r"))==NULL) return 1;

int fn=0; fn=fileno(fp);

struct stat sta;

if(fstat(fn,&sta) < 0) return 1;

printf("File size : %ld\n",(long)sta.st_size);

printf("File inode Number : %ld\n",sta.st_ino);

printf("File UID : %ld\n",(long)sta.st_uid);

printf("File GID : %ld\n", (long)sta.st_gid);
```

```
printf("File Permissions: \t");

printf( (S_ISDIR(sta.st_mode)) ? "d" : "-");

printf( (sta.st_mode & S_IRUSR) ? "r" : "-");

printf( (sta.st_mode & S_IWUSR) ? "w" : "-");

printf( (sta.st_mode & S_IXUSR) ? "x" : "-");

printf( (sta.st_mode & S_IRGRP) ? "r" : "-");

printf( (sta.st_mode & S_IWGRP) ? "w" : "-");

printf( (sta.st_mode & S_IXGRP) ? "x" : "-");

printf( (sta.st_mode & S_IROTH) ? "r" : "-");

printf( (sta.st_mode & S_IWOTH) ? "w" : "-");

printf( (sta.st_mode & S_IXOTH) ? "x" : "-");

printf("\n \n ");

printf("File type: ");

printf( (sta.st_mode & S_IRGRP) ? "r" : "-");

printf( (sta.st_mode & S_IWGRP) ? "w" : "-");

printf( (sta.st_mode & S_IXGRP) ? "x" : "-");

printf( (sta.st_mode & S_IROTH) ? "r" : "-");

printf( (sta.st_mode & S_IWOTH) ? "w" : "-");

printf( (sta.st_mode & S_IXOTH) ? "x" : "-");

printf("\n \n");

printf("File type: ");

switch (sta.st_mode & S_IFMT)
```

```
{
 case S_IFBLK: printf("block device\n"); break; case S_IFCHR:
printf("character device\n");
break;
case S_IFDIR:
printf("directory\n");
break;
case S_IFIFO:
printf("FIFO/pipe\n"); break;
case S_IFLNK:
printf("symlink\n");
break;
case S_IFREG:
printf("regular file\n");
break;
case S_IFSOCK:
printf("socket\n");
break;
default: printf("unknown? \n");
break;
}
return 0;
```

```
}
```

```
aditi@aditi-Lenovo-ideapad-330S-14IKB-U:~/ADnOR/Assignments/3B$ ./a.out
/home/aditi/ADnOR/readme
File size : 11
File inode Number : 4480231
File UID : 1000
File GID : 1000
File Permissions:       -rw-rw-r--

 File type: rw-r--

File type: regular file
```

Output:

Conclusion:

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

References:

https://www.lix.polytechnique.fr/~liberti/public/computing/prog/c/C/FUNCTIONS/stat.html