**EXP NO :13 FILE STRUCTURE**

**Aim**:

To implement the files structure to know the file properties.

**Algorithm**:

Step1: Declare the structure stat which already contains the elements like id,node

number, protection,total size in bytes ,block size,time of last access, modification and

status change.

Step2:Use the switch case to know the file type with corresponding cases.

Step3:display the I-node number by accessing structures st\_ino.

Step4:display the file size,block allocated,last status change,file access,file modification by

accessing st\_size,st\_blocks,st\_ctime,st\_atime and st\_mtime respectively.

Step5:call the exit function

**Code**:

#include <sys/types.h>

#include <sys/stat.h>

#include <time.h>

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \*argv[]) {

struct stat sb;

if (argc != 2) {

fprintf(stderr, "Usage: %s <pathname>\n", argv[0]);

exit(EXIT\_FAILURE);

}

if (stat(argv[1], &sb) == -1) {

perror("stat");

exit(EXIT\_FAILURE);

}

printf("File type: ");

switch (sb.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;

}

printf("I-node number: %ld\n", (long) sb.st\_ino);

printf("Mode: %lo (octal)\n",

(unsigned long) sb.st\_mode);

printf("Link count: %ld\n", (long) sb.st\_nlink);

printf("Ownership: UID=%ld GID=%ld\n",

(long) sb.st\_uid, (long) sb.st\_gid);

printf("Preferred I/O block size: %ld bytes\n",

(long) sb.st\_blksize);

printf("File size: %lld bytes\n",

(long long) sb.st\_size);

printf("Blocks allocated: %lld\n",

(long long) sb.st\_blocks);

printf("Last status change: %s", ctime(&sb.st\_ctime));

printf("Last file access: %s", ctime(&sb.st\_atime));

printf("Last file modification: %s", ctime(&sb.st\_mtime));

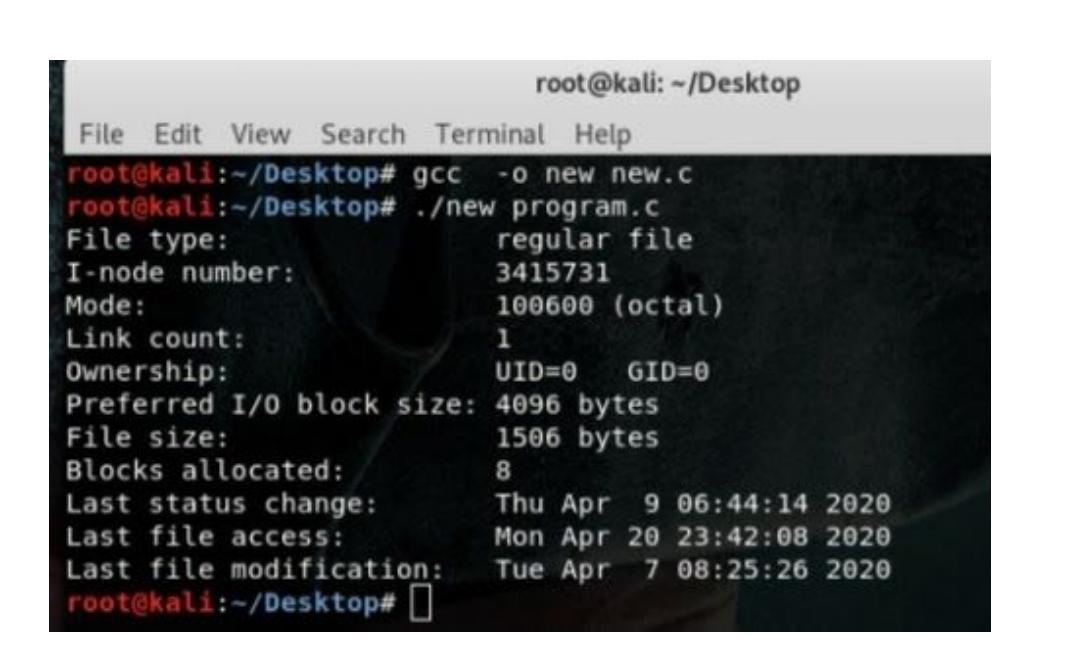
exit(EXIT\_SUCCESS);

}

**Output**:

gcc -o new new.c

./ new program.c



**Result**:

Thus the file structure is implemented.