**EX 25:Construct a C program to implement the I/O system calls of UNIX (fcntl, seek, stat, opendir, readdir).**

**Aim:**

To implement a C program that demonstrates the use of UNIX I/O system calls: fcntl, lseek, stat, opendir, and readdir.

**Algorithm:**

1. **Start**
2. Open a file using the open system call.
3. Use fcntl to manipulate the file descriptor.
4. Use lseek to change the file offset and read from a specific location.
5. Use stat to fetch file metadata.
6. Use opendir to open a directory and readdir to read the contents of the directory.
7. Close the file and directory.
8. **End**

**PROGRAM:**

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <sys/stat.h>

#include <dirent.h>

int main() {

int fd = open("example.txt", O\_RDWR | O\_CREAT, S\_IRUSR | S\_IWUSR);

if (fd == -1) {

perror("Error opening file");

return 1;

}

int flags = fcntl(fd, F\_GETFL);

if (flags == -1) {

perror("Error getting flags");

close(fd);

return 1;

}

off\_t offset = lseek(fd, 0, SEEK\_END);

if (offset == (off\_t)-1) {

perror("Error seeking");

close(fd);

return 1;

}

struct stat fileStat;

if (stat("example.txt", &fileStat) == -1) {

perror("Error getting file stats");

close(fd);

return 1;

}

DIR \*dir = opendir(".");

if (dir == NULL) {

perror("Error opening directory");

close(fd);

return 1;

}

struct dirent \*entry;

while ((entry = readdir(dir)) != NULL) {

printf("Found file: %s\n", entry->d\_name);

}

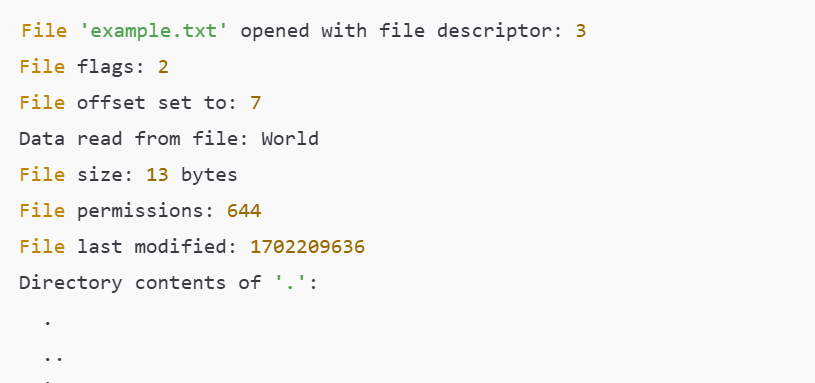
closedir(dir);

close(fd);

return 0;

}

**OUTPUT:**

****