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

Aim:

To demonstrate the use of UNIX I/O system calls, including fcntl, seek, stat, opendir, and readdir for file and directory management.

Procedure:

1. fcntl: Use the fcntl system call to manipulate the file descriptor.
2. seek: Use the lseek system call to move the file pointer to a specific location.
3. stat: Use the stat system call to retrieve information about a file.
4. opendir: Use the opendir system call to open a directory.
5. readdir: Use the readdir system call to read directory entries.

C Code:

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <sys/stat.h>

#include <dirent.h>

int main() {

int fd;

struct stat fileStat;

DIR \*dir;

struct dirent \*entry;

fd = open("example.txt", O\_CREAT | O\_WRONLY, 0644);

if (fd == -1) {

perror("Error opening file");

return 1;

}

fcntl(fd, F\_SETFL, O\_APPEND);

write(fd, "Hello, UNIX system calls!", 25);

lseek(fd, 0, SEEK\_SET);

write(fd, "Start of file", 13);

close(fd);

stat("example.txt", &fileStat);

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

dir = opendir(".");

if (dir == NULL) {

perror("Error opening directory");

return 1;

}

printf("Directory contents:\n");

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

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

}

closedir(dir);

return 0;

}

Output:

A screenshot of a computer

Description automatically generated