**EXPERIMENT – 25**

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

readdir)

#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <unistd.h>

#include <sys/stat.h>

#include <dirent.h>

#include <string.h>

#include <errno.h>

int main() {

int fd;

char buffer[100];

fd = open("sample.txt", O\_RDWR | O\_CREAT, 0644);

if (fd < 0) {

perror("open");

exit(EXIT\_FAILURE);

}

printf("File 'sample.txt' opened.\n");

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

if (flags < 0) {

perror("fcntl F\_GETFL");

} else {

printf("fcntl: File opened with flags = %d\n", flags);

}

write(fd, "ABCDEFGHIJ", 10);

lseek(fd, 5, SEEK\_SET);

read(fd, buffer, 5);

buffer[5] = '\0';

printf("lseek: Read from position 5: %s\n", buffer);

close(fd);

struct stat fileStat;

if (stat("sample.txt", &fileStat) == 0) {

printf("\nFile Information using stat():\n");

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

printf("Permissions: %o\n", fileStat.st\_mode & 0777);

printf("Last modified: %ld\n", fileStat.st\_mtime);

} else {

perror("stat");

}

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

if (dir == NULL) {

perror("opendir");

return 1;

}

struct dirent \*entry;

printf("\nFiles in current directory:\n");

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

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

}

closedir(dir);

remove("sample.txt");

return 0;

}

SAMPLE OUTPUT:

File 'sample.txt' opened.

fcntl: File opened with flags = 2

lseek: Read from position 5: FGHIJ

File Information using stat():

Size: 10 bytes

Permissions: 644

Last modified: 1719837461

Files in current directory:

- fileops

- unix\_file\_syscalls.c

- your\_other\_files...