

NAME: G. GANESH

REG NO: 192373008

EXERICSE-25

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

Aim:

To construct a C program to implement the I/O system calls of UNIX, including fcntl, seek, stat, opendir, and readdir.

Algorithm:

1. **Open File and Perform File Control (fcntl):**
 - Open a file using open.
 - Use fcntl to duplicate the file descriptor.
2. **File Positioning (seek):**
 - Use lseek to reposition the file pointer.
3. **Get File Information (stat):**
 - Use stat to fetch details about a file, such as size, permissions, and modification time.
4. **Directory Operations (opendir and readdir):**
 - Use opendir to open a directory.
 - Use readdir to list the contents of the directory.
5. **Close File and Directory:**
 - Close the opened file and directory using close and closedir.

Procedure:

- 1) Open a file using open and duplicate its descriptor using fcntl.
- 2) Use lseek to set the file pointer.
- 3) Retrieve file details using stat.
- 4) Open a directory using opendir and list its contents using readdir.
- 5) Close the file and directory.

Code:

```
#include <stdio.h>
```

```
#include <fcntl.h>

#include <unistd.h>

#include <sys/stat.h>

#include <dirent.h>

int main() {
    int fd, new_fd;
    struct stat fileStat;
    DIR *dir;
    struct dirent *entry;

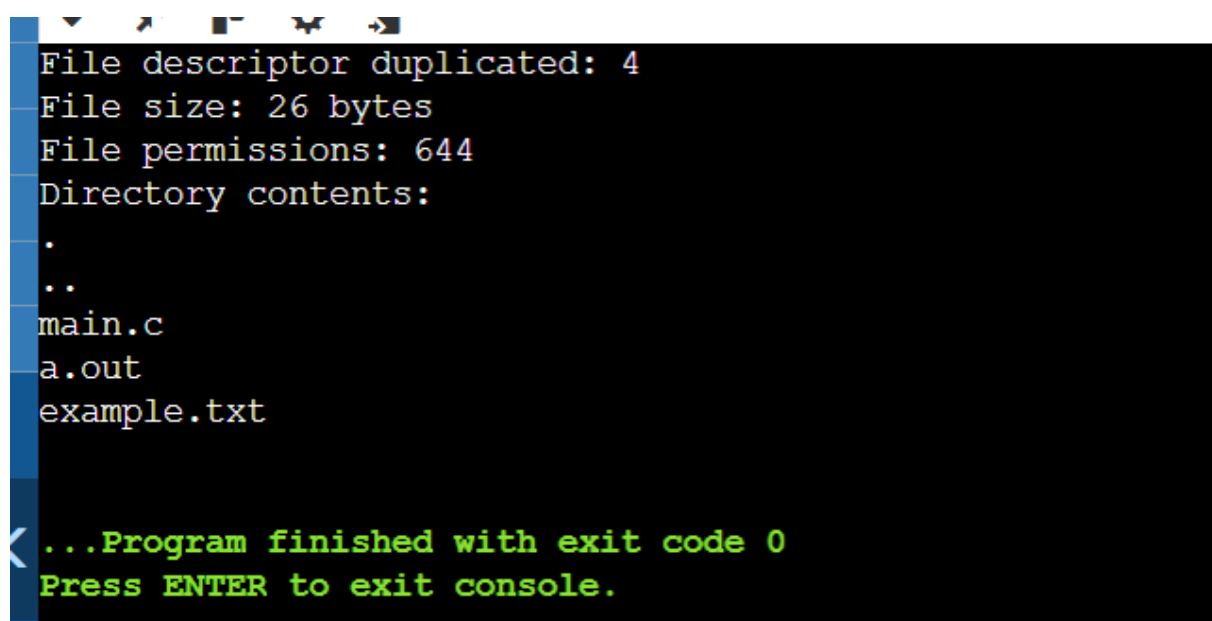
    fd = open("example.txt", O_CREAT | O_RDWR, 0644);
    if (fd < 0) {
        perror("Error opening file");
        return 1;
    }
    new_fd = fcntl(fd, F_DUPFD, 0);
    if (new_fd < 0) {
        perror("Error duplicating file descriptor");
        close(fd);
        return 1;
    }
    printf("File descriptor duplicated: %d\n", new_fd);
    lseek(fd, 0, SEEK_END);
    write(fd, "Hello, UNIX system calls!\n", 26);
    if (stat("example.txt", &fileStat) == 0) {
        printf("File size: %ld bytes\n", fileStat.st_size);
        printf("File permissions: %o\n", fileStat.st_mode & 0777);
    } else {
        perror("Error getting file stats");
    }
    dir = opendir(".");
```

```
if (dir == NULL) {
    perror("Error opening directory");
} else {
    printf("Directory contents:\n");
    while ((entry = readdir(dir)) != NULL) {
        printf("%s\n", entry->d_name);
    }
    closedir(dir);
}
close(fd);
close(new_fd);
return 0;
}
```

Result:

The program successfully demonstrates UNIX I/O system calls: `fcntl` for duplicating file descriptors, `seek` for file positioning, `stat` for fetching file details, `opendir` for opening directories, and `readdir` for listing directory contents.

Output:

A terminal window with a black background and light blue text. The output of the program is displayed line by line. At the bottom, a green prompt indicates the program has finished.

```
File descriptor duplicated: 4
File size: 26 bytes
File permissions: 644
Directory contents:
.
..
main.c
a.out
example.txt

< ...Program finished with exit code 0
Press ENTER to exit console.
```

