



# Sistemas Operativos

## Grado en Ingeniería Informática

José María Fernández

Pablo Rayón Zapater

**Práctica 5:** Gestión de ficheros y directorios



## Tabla de contenido

<b>I. Objetivos.....</b>	<b>3</b>
<b>II. Entregables.....</b>	<b>3</b>
A. Ejercicio 1.....	3
B. Ejercicio 2.....	3
C. Ejercicio 3.....	3
<b>III. Conclusiones.....</b>	<b>3</b>

## I. Objetivos

- Entender la gestión de archivos y directorios.
- Estudiar las funciones aportadas en este documento.

## II. Entregables

### A. Ejercicio 1

Crear un programa que lea veinte caracteres, a partir de la posición 50, de un fichero ya existente con reserva de memoria dinámica para la lectura del el resto del fichero.

```
1  #include <string.h>
2  #include <stdio.h>
3  #include <fcntl.h>
4  #include <stdlib.h>
5
6  main(int argc, char *argv[])
7  {
8      const char* cadena = malloc(sizeof(char)*20);
9
10     int fichero = open("file", O_CREAT | O_RDONLY, 0644);
11
12     if (fichero == -1)
13     {
14         perror("Error al abrir fichero");
15         exit(1);
16     }
17     pread(fichero, cadena, 20, 50);
18     printf("%s", cadena);
19
20     close(fichero);
21     return 0;
22 }
```

En este ejercicio, realizamos en nuestro array "cadena", una asignación dinámica de memoria de tamaño 20. Después abrimos un fichero el cual solo tiene permisos de lectura. Comprobamos que se puede abrir el fichero y lo leemos desde la posición 50 por medio de la función pread, la cual nos permite leer un fichero a partir de un offset, en este caso 50, y lo almacenamos en nuestro array dinámico de caracteres. Para finalizar, imprimimos el array.

## B. Ejercicio 2

Crear un programa que copie el contenido de un archivo “origen” hacia un archivo “destino” y que se muestre por pantalla el contenido copiado.

En este ejercicio, abrimos un archivo origen el cual solo tiene permiso de lectura y otro a el cual llamamos destino, que solo tiene carácter de escritura. Si no los podemos abrir, acabara el programa.

```
1  #include <string.h>
2  #include <fcntl.h>
3  #include <stdlib.h>
4  #include <stdio.h>
5
6  char aux[];
7  int main(int argc, char *argv[])
8  {
9      int origen = open("origen", O_RDONLY | O_CREAT, 0644);
10     int destino = open("destino", O_WRONLY | O_CREAT, 0644);
11     int size = 0;
12     if (origen == -1 || destino == -1)
13     {
14         perror("Error al abrir fichero");
15         exit(1);
16     }
17     while (read(origen, aux, 1) != 0)
18     {
19         printf("%s", aux);
20         write(destino, aux, 1);
21     }
22     close(origen);
23     close(destino);
24 }
```

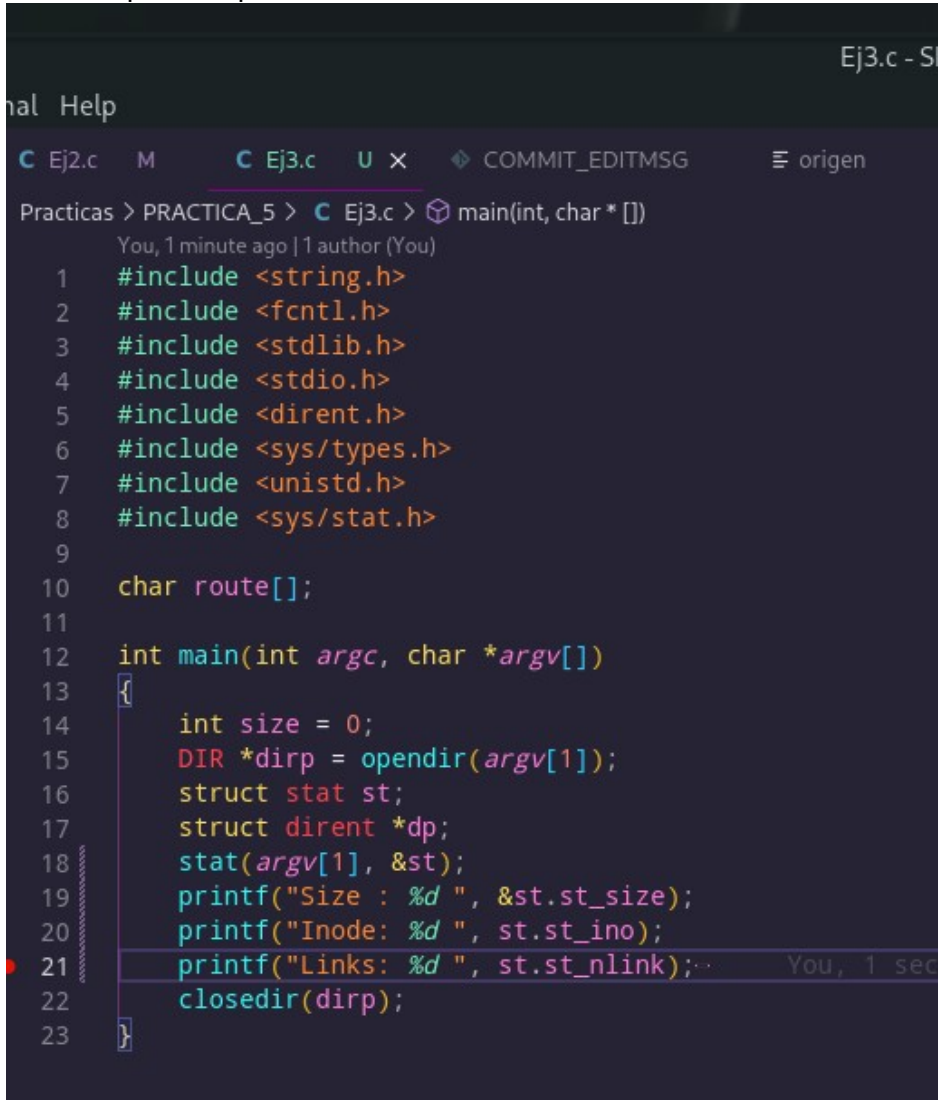
## C. Ejercicio 3

Escribir un programa que calcule la suma de los bytes ocupados por todos los ficheros y directorios que estén contenidos a partir de un directorio dado como parámetro. Además, mostrar por pantalla los siguientes parámetros:

- File Size
- Number of Links
- File inode
- File Permissions
- Symbolic link (status): is or is not

¿Qué ocurre cuando hay dos enlaces hard que hacen referencia al mismo fichero?  
**Justificar el proceso 4 PUNTOS.**

No se ha conseguido realizar satisfactoriamente este ejercicio, se ha conseguido leer la cantidad de espacio libre que queda en este directorio, el numero de inodo y el numero de hard links que este posee.



```
Ej3.c - S
hal Help
C Ej2.c M C Ej3.c U x COMMIT_EDITMSG origen
Practicas > PRACTICA_5 > C Ej3.c > main(int, char *[])
You, 1 minute ago | 1 author (You)
1 #include <string.h>
2 #include <fcntl.h>
3 #include <stdlib.h>
4 #include <stdio.h>
5 #include <dirent.h>
6 #include <sys/types.h>
7 #include <unistd.h>
8 #include <sys/stat.h>
9
10 char route[];
11
12 int main(int argc, char *argv[])
13 {
14     int size = 0;
15     DIR *dirp = opendir(argv[1]);
16     struct stat st;
17     struct dirent *dp;
18     stat(argv[1], &st);
19     printf("Size : %d ", &st.st_size);
20     printf("Inode: %d ", st.st_ino);
21     printf("Links: %d ", st.st_nlink);
22     closedir(dirp);
23 }
```

Un enlace hard al ser una referencia que apunta a un archivo no tiene porque entrar en conflicto con otro que apunte a el mismo, lo único es que al verse modificado el archivo ambos enlaces deberán ser modificados