

Introducción a la Programación en C

–Gestión de Ficheros–

Christopher Expósito-Izquierdo
cexposit@ull.edu.es

Airam Expósito-Márquez
aexposim@ull.edu.es

Israel López-Plata
ilopezpl@ull.edu.es

Belén Melián-Batista
mbmelian@ull.edu.es

José Marcos Moreno-Vega
jmmoreno@ull.edu.es



Contenidos

- 1 Introducción
- 2 Archivos
- 3 FILE
- 4 Apertura y Cierre
- 5 Lectura y Escritura

Introducción:

- La información almacenada en la memoria del ordenador es **volátil**. Esto significa que cuando un programa finaliza su ejecución, el contenido de la memoria se pierde
- Los ordenadores emplean medios de **almacenamiento secundario** para almacenar información de forma duradera
- Los medios secundarios tienen la ventaja de que **la información no se pierde** cuando se apaga el ordenador
- La información en los medios secundarios se almacena en **archivos**

Archivos:

Archivo

Es una agrupación lógica de datos, generalmente contigua, a la que se accede en un cierto orden

- Los archivos son leídos en el lenguaje de programación C por líneas de caracteres
- Cada línea de caracteres está formada por cero o más caracteres terminados con un carácter de salto de línea
- La última línea puede no terminar en un salto de línea

FILE:

- **FILE** es un tipo de objeto capaz de almacenar información necesaria para gestionar un archivo
- Un **FILE** tiene los siguientes elementos:
 - Indicador de posición en el archivo
 - Indicador de error que indica si se ha producido un error de E/S
 - Indicador de fin de archivo (End of File: EOF) que indica si se ha alcanzado el fin del archivo
 - ...

Apertura y Cierre:

La forma de abrir un archivo es tal como sigue:

```
FILE* file = fopen(nombreArchivo, modo)
```

donde:

- **nombreArchivo**: nombre del archivo a abrir
- **modo**: especifica la forma de apertura del archivo

Modo	Significado
r	Abre un archivo existente para lectura
w	Abre un nuevo archivo para escritura. Si existe el archivo se borra su contenido. Si no existe se crea
a	Abre un archivo existente para añadir datos al final. Si no existe se crea
r+	Abre un archivo existente para lectura y escritura
w+	Abre un archivo nuevo para escritura y lectura. Si existe lo borra. Si no existe lo crea
a+	Abre un archivo para leer y añadir

Apertura y Cierre:

- Si `fopen` devuelve `NULL` el archivo no se pudo abrir, lo que indica un error
- Las causas de los fallos son variadas, pero muchas están relacionadas con el modo de apertura solicitado
 - Si se abre un archivo con modo `r`, se obtendrá un fallo si el archivo no existe o no se puede leer
 - Otra causa de errores es intentar abrir más archivos del permitido (ver macro `FOPEN_MAX` de `stdio.h`)

Apertura y Cierre:

Cualquier archivo abierto se puede cerrar mediante la función `fclose`:

```
fclose(file)
```

donde

- `file`: es el archivo a cerrar

`fclose` devuelve un entero indicando si el archivo se cerró correctamente o no:

- `0`: el archivo se cerró correctamente
- `≠ 0`: se ha producido algún error durante el cierre del archivo


```
#include <stdio.h>

void main() {
    FILE* file = fopen("myFile.txt", "r");
    if (file == NULL) {
        printf("There was an error when opening\n");
    } else {
        printf("The file has been open correctly\n");
        fclose(file);
    }
}
```

Lectura y Escritura:

- Las operaciones de E/S a archivos descritas en el lenguaje de programación C permiten acceder a los datos de los archivos con un cierto formato, que puede ser definido por el usuario
- Las principales operaciones de E/S con formato son la lectura, a través de la función `fscanf` y las de escritura, a través de la función `fprintf`

Lectura y Escritura:

La función `fscanf` permite leer datos desde un archivo. Se define tal como sigue:

```
numeroElementoLeidos fscanf(archivo, formato,...);
```

donde

- **numeroElementoLeidos**: número de elementos leídos desde el archivo. Este número es negativo si se ha producido un error durante la lectura. En el caso de que el error se produzca antes de la conversión de formatos, toma el valor **EOF**
- **archivo**: archivo desde el que se quiere leer
- **formato**: secuencia de caracteres que indica los elementos a leer
- Los restantes parámetros son las variables en donde se almacenarán los elementos leídos

```
#include <stdio.h>

void main() {
    int rows;
    int columns;
    FILE* file = fopen("myFile.txt", "r");
    if (file == NULL) {
        printf("There was an error when opening\n");
    } else {
        int read = fscanf(file, "%i %i", &rows, &columns);
        printf("Elements read from file: %i\n", read);
        if (read != 2) {
            printf("There was an error when reading\n");
        } else {
            printf("\tRows: %i\n", rows);
            printf("\tColumns: %i\n", columns);
        }
        fclose(file);
    }
}
```

Lectura y Escritura:

La función `fprintf` permite escribir datos en un archivo. Se define tal como sigue:

```
numeroElementoEscritos fprintf(archivo, formato,...);
```

donde

- **numeroElementoEscritos**: número de elementos escritos en el archivo. Este número es negativo si se ha producido un error durante la escritura
- **archivo**: archivo en el que se quiere escribir
- **formato**: secuencia de caracteres que indica los elementos a escribir
- Los restantes parámetros son las variables a escribir

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

```
void main() {  
    int rows;  
    int columns;  
    FILE* fileIn = fopen("myFile.txt", "r");  
    fscanf(fileIn, " %i %i", &rows, &columns);  
    printf("Elements read from file:\n");  
    printf("\tRows: %i\n", rows);  
    printf("\tColumns: %i\n", columns);  
    fclose(fileIn);  
    FILE* fileOut = fopen("myFileOut.txt", "w");  
    fprintf(fileOut, " %i %i", rows, columns);  
    printf("Data have been written successfully\n");  
    fclose(fileOut);  
}
```

Lectura y Escritura:

Es posible comprobar si se ha alcanzado el final de un archivo o si se ha producido un error durante las operaciones de E/S

- **feof(file)** comprueba si se ha alcanzado EOF. En caso negativo devuelve 0, devolviendo un valor no nulo si se ha llegado al final del archivo
- **ferror(file)** comprueba si se ha generado un error en las operaciones de E/S sobre un archivo. Si no existe error, devuelve 0. En caso contrario devuelve un valor no nulo

```
#include <stdio.h>

void main() {
    int number1;
    int number2;
    FILE* file = fopen("myFile.txt", "r");
    do {
        int read = fscanf(file, "%i %i", &number1, &number2);
        if (read == 2) {
            printf("Elements read from file:\n");
            printf("\tNumber 1: %i\n", number1);
            printf("\tNumber 2: %i\n", number2);
        }
    } while (!feof(file) && !ferror(file));
    fclose(file);
}
```


Introducción a la Programación en C –Gestión de Ficheros–

Christopher Expósito-Izquierdo
cexposit@ull.edu.es

Airam Expósito-Márquez
aexposim@ull.edu.es

Israel López-Plata
ilopezpl@ull.edu.es

Belén Melián-Batista
mbmelian@ull.edu.es

José Marcos Moreno-Vega
jmmoreno@ull.edu.es

