

## Gestión de Ficheros en C

- Definición:

`FILE *FicheroLogico; //Declara un puntero a una variable fichero lógico.`

- Apertura y creación:

Apertura de un fichero en distintos modos. Si hay algún error devuelve NULL.

Formato: `FILE * fopen ( const char * filename, const char * mode );`

Donde:

- filename es una cadena que debe contener el nombre o la dirección dentro del disco donde abrir o crear el fichero.
- mode es una cadena que indica cómo abrir y crear el fichero.

"r"	<b>leer:</b> Abre el fichero en modo texto para solo leer. El fichero debe existir.
"rb"	<b>Igual que "r" pero para ficheros binarios.</b>
"w"	<b>Escribir:</b> Crea un fichero en modo texto para solo escribir, si existe lo borra previamente.
"wb"	<b>Igual que "w" pero para ficheros binarios.</b>
"a"	<b>Solo añadir:</b> Abre un fichero en modo texto para solo escribir al final del mismo. Si el fichero no existe lo crea.
"ab"	<b>Igual que "a" pero para ficheros binarios.</b>
"r+"	<b>Lectura/Escritura:</b> Abre un fichero en modo texto para leer y escribir. El fichero debe existir previamente.
"r+b"	<b>Igual que "r+" pero para ficheros binarios.</b>
"w+"	<b>Escritura/Lectura:</b> Igual que el modo "r+" pero en caso de existir lo borra previamente.
"w+b"	<b>Igual que "w+" pero para ficheros binarios.</b>
"a+"	<b>Solo añadir/Lectura:</b> Abre un fichero en modo texto para solo escribir al final del mismo aunque se puede leer de otras partes del fichero. Si el fichero no existe es creado.
"a+b"	<b>Igual que "a+" pero con fichero binarios.</b>

`FILE *fdatos=fopen("datos.dat","r+b"); //Abre un fichero para leer y escribir en modo  
// binario. Debe existir previamente.`

`FILE *fresult=fopen("salida.dat","w+b"); //Crea un fichero para leer y escribir en modo  
// binario. Si existe un fichero previamente lo borra.`

- Cerrar un fichero:

Cierra un fichero. Si todo ha ido bien devuelve un 0

Formato: `int fclose ( FILE * stream );`

`fclose(fdatos); //cierra el fichero lógico fdatos abierto previamente.`

- Acceso directo en ficheros:

Mueve el puntero del fichero lógico (stream) un número de bytes (offset) desde una posición determinada (origin)

Formato: `int fseek ( FILE * stream, long int offset, int origin );`

La posiciones (origin) son:

SEEK_SET	Desde el principio del fichero hasta el final
SEEK_CUR	Desde la posición actual del puntero de fichero
SEEK_END	Desde el final del puntero hasta el principio

```
fseek(fdatos, sizeof(float)*10, SEEK_SET); //Se desplaza 10 reales desde el principio del
//fichero antes de leer del fichero lógico fdatos.
```

```
fseek(fdatos,0,SEEK_END); //Se desplaza al final del fichero lógico fdatos
```

Devuelve el número de bytes que se ha desplazado el puntero lógico tras un fseek.

Formato: **long int ftell ( FILE \* stream );**

```
FILE *fdatos=fopen("datos.dat","r+b");
fseek(fdatos,0,SEEK_END);
long Ndatos=(float)ftell(fdatos)/sizeof(float); //Número de reales dentro del fichero.
fclose(fdatos);
```

- Lectura y escritura:

Lee desde el fichero un número de bytes (size), un número de veces (count) y lo deja memoria apuntada por ptr. Devuelve el número de elementos leídos, si no ha habido errores devuelve el mismo valor que count.

Formato: **size\_t fread ( void \* ptr, size\_t size, size\_t count, FILE \* stream );**

```
int codigo;
Cliente C;
float puntuaciones[100];
//Suponiendo que el fichero tiene un entero, una estructura y un vector de reales
FILE *fdatos=fopen("datos.dat","r+b");
fread(&codigo, sizeof(codigo),1,fdatos);
fread(&C, sizeof(Cliente),1,fdatos);
fread(puntuaciones, sizeof(float),100,fdatos); //Ojo con los vectores no se utiliza &
fclose(fdatos);
```

Escribe en el fichero un número de bytes (size), un número de veces (count) los datos apuntados por ptr. Devuelve el número de elementos escritos, si no ha habido errores devolverá el mismo valor que.

Formato: **size\_t fwrite ( void \* ptr, size\_t size, size\_t count, FILE \* stream );**

```
int codigo;
Cliente C;
float puntuaciones[100];
//Suponiendo que las variables tienen datos correctos se guardaría del siguiente modo:
FILE *fdatos=fopen("datos.dat","w+b");
fwrite(&codigo, sizeof(codigo),1,fdatos);
fwrite(&C, sizeof(Cliente),1,fdatos);
fwrite(puntuaciones, sizeof(float),100,fdatos); //Ojo con los vectores no se utiliza &
fclose(fdatos);
```

Devuelve EOF si ha leído la marca de fin de fichero.

Formato: **int feof ( FILE \* stream );**

//Suponiendo que tenemos un fichero lleno de reales, procedemos a sumarlos.

float suma=0, dato;

FILE \*fiche=fopen("datos.dat","r+b");

fread(&dato, sizeof(dato),1,fiche);

while (feof(fiche) != EOF)

{

    Suma += dato;

    fread(&dato, sizeof(dato),1,fiche);

}

fclose(fiche);