Tema 6:

Ficheros

Ficheros

- **Archivo, o fichero:**
  - Es una estructura de datos en memoria externa, cuyo **tiempo de vida no** está ligado a la ejecución del programa que lo crea o lo maneja.
  - Un fichero existe desde que un programa lo crea y **permanece** mientras no sea destruido por este o por otro programa.

- **Características:**
  - Pueden almacenar gran cantidad de información.
  - La información permanece a los procesos que lo generan y utilizan.
  - Pueden ser accedidos desde diferentes procesos o programas.

Ficheros

- Existen tres ficheros predefinidos en C:
  - stdin (entrada estándar: teclado)
  - stdout (salida estándar: pantalla)
  - stderr (salida estándar para mensajes de error)

- Para trabajar con ficheros en C: las operaciones de entrada y salida, los tipos de dato y constantes predefinidas están recogidas en la biblioteca **stdio.h**.

Tipos de Ficheros

Ficheros de texto y ficheros binarios:

- En un **fichero de texto**:
  - los bytes se interpretan como caracteres.
  - toda la información que se puede guardar en un fichero de texto son caracteres (se pueden interpretar como valores numéricos).
  - esa información podrá ser visualizada por un editor de texto.

- En un **fichero binario**:
  - los datos se almacenan en formato binario (según la codificación interna del ordenador).
  - los ficheros binarios no pueden visualizarse por un editor de texto.

Ficheros. Declaración

- La sintaxis para la declaración de un fichero es:

FILE *nombre_fichero;

Ejemplo:

```
FILE  *miFichero;
FILE  *fichero;
FILE  *fe, *fs;
```

// miFichero, fichero, fe y fs son punteros a ficheros

Ficheros. Apertura

- La primera operación que hay que hacer con un fichero es abrirlo, es decir, indicarle al sistema operativo donde se encuentra físicamente y como se accede a los datos.
- La sintaxis es:

nombre_fichero= fopen ("fichero_fisico", modo)

- Siendo:
  - **fichero_fisico**: es el nombre del fichero almacenado en el dispositivo externo, debe ir con el camino.
  - **modo**: es el modo de acceso.

Ficheros. Apertura

- Los **modos de acceso** que vamos a utilizar son:

"r" **Abre** un fichero de texto para **lectura**.
El fichero debe **existir**.

"w" **Abre** un fichero de texto para **escritura**.
Si ese fichero **no existe, lo crea**.
Si **existe** el fichero, **se vuelve a escribir** desde el principio, perdiéndose la información que tenía

"a" **Abre** un fichero de texto para **escritura**.
Si ese fichero **no existe, lo crea**.
Si **existe** el fichero, los datos nuevos **se añaden al final** del fichero.

Ejemplos

```
FILE  *mifichero;

mifichero = fopen (“c:\\entrada.txt”, “r”);

if (mifichero==NULL)
       printf(“No se accede al fichero”);
else {
 /* el fichero se ha abierto correctamente */
}
```

fopen **abre** el fichero entrada.txt para **lectura**.

Si no se hubiera podido abrir el fichero devuelve valor **NULL**

Ejemplos

```
FILE *fichero;
fichero = fopen("c:\\datos.dat", "w");
```

Abre el fichero *datos.dat* para **escritura**.
Si ese fichero ya existía, se crea otro nuevo vacío.

```
FILE *fe;
fe = fopen ("c:\\datos.dat", "a");
```

Abre el fichero *datos.dat* para escritura **añadiendo** nuevos valores **a partir del final** del fichero.

Ficheros. Cierre

- **Cierre del fichero abierto.** La función ***fclose*** cierra el fichero que ha sido abierto mediante *fopen*.

- La sintaxis es:

fclose (nombre_fichero)

No hacer el cierre de un fichero puede generar problemas, entre ellos la pérdida de información del fichero.

Ejemplos:

```
fclose (mifichero);
fclose (FE);
fclose (FS);
```

Ficheros. Lectura y escritura

- Las funciones ***fscanf*** y ***fprintf*** de entrada y salida de datos por fichero tienen un uso semejante a las funciones *scanf* y *printf*, de entrada y salida por consola.
- Su sintaxis es:

fscanf (nomb_fich, cad_formato , &var1, &var2, ..., &varn);

fprintf (nomb_fich, cad_formato, expres1, expres2,..., expresn);

nomb_fichero es un fichero abierto en formato de tipo texto

La función ***fscanf*** devuelve un valor entero. Si se intenta hacer una lectura y no hay datos en el fichero devuelve un valor entero especial, el valor **EOF** (End Of File)

Procesar el fichero por líneas

Podemos hacer recorridos de la información almacenada en un fichero:

Por líneas, utilizando '\n' como centinela:

Las líneas de un fichero de texto finalizan con el carácter '\n', por tanto podríamos utilizarlo como centinela para recorrer una línea

```
fscanf( /*lee valor para dato caracter*/ );

while (dato!= '\n')
{
   /*procesar dato*/
   fscanf( /*lee valor para dato caracter*/ );
}
/* observar que la lectura se hace al final del ciclo */
```

Procesar el fichero completo

Para **recorrer todo el fichero** podemos hacer varias cosas:

- Utilizar la constante especial: **`EOF`**
  - EOF (End Of File)
  - Cuando scanf o fscanf encuentra el final del fichero (no pudiendo almacenar datos en las variables)
  - Se puede utilizar de centinela

- Utilizar la función fin fichero: **`foef(nombre_fich)`**
  - La función **feof** devuelve un valor distinto de 0 (cierto) cuando se lee el valor fin de fichero y 0 en otro caso

EOF y Ficheros de texto. Esquema

Recorrido de un fichero de texto controlado por EOF:

```
int fin;
fin= fscanf( fich, /*lee valores para variables*/ );
while (fin!=EOF)
{
    /*procesar valores leidos*/
    fin= fscanf( fich, /*lee valores para variables*/ );
 }
```

EOF y Ficheros de texto. Ejemplo

Programa que lee un fichero de números enteros **FN** y escribe los números pares en un fichero **FP**.

```
#include <stdio.h>
int main ()
{
      int num, fin;
      FILE *FN;
      FILE *FP;
      FN = fopen ("fNumeros.txt","r");
      FP = fopen ("fPares.txt","w");

      fin= fscanf(FN,"%d", &num);
      while (fin != EOF ) {
            if (num%2==0)
            fprintf (FP,"%d ", num);
            fin= fscanf(FN,"%d", &num);
      }
      fclose(FN);
      fclose(FP);
      return 0;
}
```

feof y Ficheros de texto. Esquema

Recorrido de un fichero de texto controlado por feof():

```
fscanf( fich, /*lee valores para variables*/ );
while (!feof (fich))
{
      /*procesar valores leidos*/
      fscanf(fich,  /*lee valores para variables*/ );
}
```

feof y Ficheros de texto. Ejemplo

Programa que lee un fichero de números enteros **FN** y escribe los números pares en un fichero **FP**.

```
#include <stdio.h>
int main ()
{
      int num;
      FILE *FN;
      FILE *FP;
      FN = fopen ("fNumeros.txt","r");
      FP = fopen ("fPares.txt","w");

      fscanf(FN,"%d", &num);
      while (!feof (FN)) {
          if (num%2==0)
             fprintf (FP,"%d ", num);
          fscanf(FN,"%d", &num);
      }
      fclose(FN);
      fclose(FP);
      return 0;
}
```