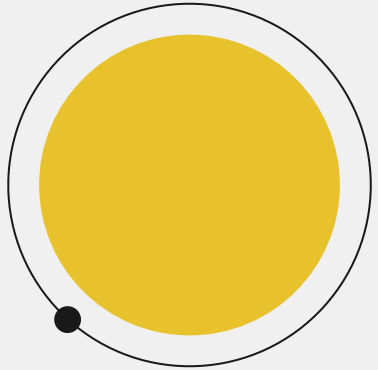
Archivos

¿Qué es?

Las grandes cantidades de datos se almacenan normalmente en un dispositivo de memoria secundaria. Estas colecciones de datos se conocen como archivos.

Un archivo es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son de igual tipo y constan a su vez de diferentes entidades de nivel más bajos denominadas campos.



Hay dos tipos de archivos

Texto (.txt)

Un archivo de texto es una secuencia de caracteres organizadas en líneas terminadas por un carácter de nueva línea.

• En estos archivos se pueden almacenar, fuentes de programas, texto plano, base de datos simples, etc.

• Los archivos de texto se caracterizan por ser planos, es decir, solo contienen caracteres de texto.

Binario (.dat)

Es una secuencia de bytes que tienen una correspondencia uno a uno con un dispositivo externo. Así que no tendrá lugar ninguna traducción de caracteres.

• El número de bytes escritos (leídos) será el mismo que los encontrados en el dispositivo externo.
• E.g. de estos archivos son Fotografías, imágenes, texto con formatos, archivos ejecutables (aplicaciones), etc.

- Se puede conseguir la entrada y la salida de datos a un archivo a través del uso de la biblioteca de funciones estándar "stdio.h"; C puro no tiene palabras claves que realicen las operaciones de E/S.

- La tabla siguiente da un breve resumen de las funciones que se pueden utilizar. Observe que la mayoría de las funciones comienzan con la letra “f” (file).

Nombre	Función
fopen()	Abre un archivo
fclose()	Cierra un archivo
fgets()	Lee una cadena de un archivo
fputs()	Escribe una cadena de un archivo
fseek()	Busca un byte especifico de un archivo
fprintf()	Escribe una salida con formato en el archivo
fscanf()	Lee una entrada con un formato en el archivo
feof()	Devuelve cierto si se llega al final del archivo
ferror()	Devuelve cierto si se produce un error
rewind()	Coloca el cursor de posición en el archivo al principio del mismo
remove()	Borra un archivo
fflush()	Vacía un archivo
fread()	Lee un bloque de una “stream” de datos (binario)
fwrite()	Escribe un bloque de datos a un archivo como “stream” (binario)

El uso de punteros en Archivos

- Un puntero a un archivo es un puntero a una información que define varias cosas sobre él, incluyendo el nombre, el estado y la posición actual del archivo.

- Un puntero a un archivo es una variable de tipo puntero al tipo FILE que se define en "stdio.h".

- Un programa necesita utilizar punteros a archivos para leer o escribir en los mismos. Para obtener una variable de este tipo se utiliza una secuencia como esta: **FILE *F**

01

Apertura de Archivos

- La función **fopen()** abre una secuencia para que pueda ser utilizada y la asocia a un archivo.

  Su prototipo es: **FILE * fopen (const char nombre_archivo, const char modo);**

- Donde **nombre_archivo** es un puntero a una cadena de caracteres que representan un nombre valido del archivo y puede incluir una especificación del directorio.

- La cadena a la que apunta **modo** determina como se abre el archivo

Modos de apertura de un archivo

Modo	Significado
r	Abre un archivo de texto para lectura *(read)*
w	Abre un archivo de texto para escritura desde el comienzo *(write)*
a	Abre un archivo de texto para escritura al final del archivo *(append)*
rb	Abre un archivo binario para lectura *(read binary)*
wb	Abre un archivo de binario para escritura desde el comienzo *(write binary)*
ab	Abre un archivo binario para escritura al final del archivo *(append binary)*
r+	Abre o crea un archivo de texto para lectura / escritura
w+	Crea un archivo de texto para lectura / escritura
a+	Añade o crea un archivo de texto para lectura / escritura
rb+	Abre o crea un archivo binario para lectura / escritura
wb+	Crea un archivo binario para lectura / escritura
ab+	Añade o crea un archivo binario para lectura / escritura

- Si se produce un error cuando se esta intentando abrir un archivo, fopen() devuelve un puntero nulo (NULL).
- Si se usa fopen() para abrir un archivo para escritura, entonces cualquier archivo existente con el mismo nombre se borrará y se crea uno nuevo
- Si se quiere añadir al final del archivo entonces debe usar el modo a. Si se usa a y no existe el archivo, se devolverá un error.
- La apertura de un archivo para las operaciones de lectura requiere que exista el archivo. Si no existe, fopen() devolverá un error.

```c
#include <stdio.h>
int main ()
{
    FILE * archivo = fopen("miarchivo.txt", "w");
    if(archivo == NULL){
        printf("Error en la apertura del archivo");
        return 1;
        }
    fprintf(archivo,"Hola estoy ecrbiendo en el archivo");
    fprintf(archivo,"Esta es la segunda linea de mi archivo");

     printf("Tu archivo a sido creado");

     fclose(archivo);

    return 0;
}
```

Manejo de archivos en C (Modo texto)

- Para introducir u obtener datos en modo texto de un archivo tenemos las siguientes cuatro funciones:
  • fprintf()
  • fscanf()
  • fgets()
  • fputs()
- Estas funciones se comportan exactamente como prinft() y scanf(), gets() y puts() excepto que operan sobre archivo. Sus prototipos son:

```
int fprintf(FILE *F, const char *cadena_de_control, .....);
int fscanf(FILE *F, const char *cadena_de_control, .....);
char *fputs(char *str, FILE *F);
char *fgets(char *str, int long, FILE *F);
```

- El operar con los archivos en modo texto facilita la comprensión de los archivos por cualquier usuario que logre abrir el archivo con otra aplicación que permita leer el texto claro.