LABORATORIO DE COMPUTACIÓN 1

Unidad 6 – Archivos

Tema: Archivos de texto

ARCHIVOS

En **informática**, se conoce como **archivo** o fichero a un conjunto organizado de unidades de información (bits) almacenados en un dispositivo.

El objetivo es almacenar información en un medio persistente, y recuperar la información grabada previamente. Esta información se almacena con dos formatos muy distintos desde el punto de vista de su manipulación: el formato de texto y el formato binario.

TIPOS DE ARCHIVOS

ARCHIVO DE TEXTO

- Es una secuencia de caracteres organizadas en líneas terminadas por un carácter de nueva línea.
- Los archivos de texto se caracterizan por ser planos, es decir, todas las letras tienen el mismo formato y no hay palabras subrayadas, en negrita o letras de distinto tamaño o ancho.
- Ejemplos: textos en general, base de datos simples, entre otros.

ARCHIVO BINARIO

- Contiene información codificada en lenguaje binario (el contenido se trata de una secuencia de bytes que se agrupan de ocho en ocho)
- No tendrá lugar ninguna traducción de caracteres.
- El número de bytes escritos (leídos) será el mismo que los encontrados en el dispositivo externo.
- Ejemplos: fotografías, imágenes, texto con formatos, archivos ejecutables (aplicaciones), entre otros.

FUNCIONES PARA EL MANEJO DE ARCHIVOS

Se debe incluir la librería STDIO.H.

NOMBRE	FUNCION
fopen()	Abre un archivo.
fclose()	Cierra un archivo.
fgets()	Lee una cadena de un archivo.
fputs()	Escribe una cadena en un archivo
fseek()	Busca un byte específico de un archivo.
fprintf()	Escribe una salida con formato en el archivo
fscanf()	Lee una entrada con formato desde el archivo.
feof()	Devuelve cierto si se llega al final del archivo.
ferror()	Devuelve cierto si se produce un error.
rewind()	Coloca el localizador de posición del archivo al principio del mismo.
remove()	Borra un archivo.
fflush()	Vacía un archivo.

MANEJO DE ARCHIVOS

EL PUNTERO A UN ARCHIVO

- Es una variable de tipo puntero a la estructura FILE que se define en STDIO.H.
- Es un puntero a una información que define el nombre, el estado y la posición actual del archivo.
- Para usar un archivo tenemos que asociar una variable puntero * al archivo para poder escribir o leer de el.
- Para obtener una variable de este tipo se utiliza una secuencia como ésta:

```
FILE *F;
```

MANEJO DE ARCHIVOS

APERTURA DE UN ARCHIVO

La función que realiza la operación de apertura de un archivo se denomina **fopen()**

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

Donde:
nombre_archivo: nombre válido del archivo al que se quiere acceder para leer o escribir y puede incluir una especificación del directorio.
modo: determina cómo se abre el archivo (únicamente para leer, únicamente para escribir, para ambas, etc.).

Si se produce un error cuando se está intentando abrir un archivo, fopen() devuelve un puntero nulo.

MANEJO DE ARCHIVOS

MODOS DE APERTURA DE UN ARCHIVO

Modo	Significado
r	Abre un archivo de texto para lectura
w	Crea un archivo de texto para escritura.
a	Abre un archivo de texto para añadir.
rb	Abre un archivo binario para lectura.
wb	Crea un archivo binario para escritura.
ab	Abre un archivo binario para añadir.
r+	Abre 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.
r+b	Abre un archivo binario para lectura / escritura
w+b	Crea un archivo binario para lectura / escritura.
a+b	Añade o crea un archivo binario para lectura / escritura.

MANEJO DE ARCHIVOS

CIERRE DE UN ARCHIVO

La función fclose() cierra una secuencia que fue abierta mediante una llamada a fopen().

Su prototipo es:
int fclose(FILE *F);

Escribe toda la información que todavía se encuentre en el buffer en el disco y realiza un cierre formal del archivo a nivel del sistema operativo.

Un error en el cierre de una secuencia puede generar todo tipo de problemas, incluyendo la pérdida de datos, destrucción de archivos y posibles errores intermitentes en el programa.

MANEJO DE ARCHIVOS

LECTURA Y ESCRITURA DE UN ARCHIVO

Las funciones fprintf() y fscanf() se comportan exactamente como las funciones prinft() y scanf() (para imprimir y leer en y desde la consola), 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, ...);

Donde F es un puntero al archivo devuelto por una llamada a fopen().

MANEJO DE ARCHIVOS

LECTURA Y ESCRITURA DE UN ARCHIVO

Las funciones fgets() y fputs() pueden leer y escribir cadenas hacia o desde los archivos.

Sus prototipos son:

char *fputs(char *str, FILE *F);

char *fgets(char *str, int long, FILE *F);