

Programación II

Archivos

Archivos

DEFINICIÓN

Un archivo puede definirse como un conjunto de información relacionada, almacenada en un dispositivo externo - generalmente un disco u otro dispositivo de almacenamiento masivo-, que tiene un ***nombre*** para su identificación, y una ***ubicación*** -ruta- dentro del dispositivo.



Archivos de sistema



Archivos ejecutables



Archivos de datos

Archivos

CLASIFICACIÓN

Archivos de sistema: son aquellos utilizados por el sistema operativo para todas las funciones de administración de la máquina.

Archivos ejecutables: son los programas o aplicaciones que permiten a los usuarios realizar tareas concretas, tales como un documento de texto, un navegador, u otro programa. La única acción posible sobre este tipo de archivos es su ejecución y uso.

Archivos

CLASIFICACIÓN

Archivos de datos: son los archivos que el operador crea a partir de una aplicación, como un procesador de texto, un editor de imágenes, una planilla de cálculo, etc. Cada una de estas aplicaciones permite la manipulación de archivos específicos (creación, modificación, asignación de nombre y ruta, etc.).

Archivos de datos

Datos con un formato y/o estructura específica que son interpretados por programas para poder crearlos, visualizarlos, editarlos y/o borrarlos.



Archivos de datos



Planilla de cálculo



Archivo PDF



Código fuente



Archivo de clientes

Archivos de datos

Un registro de información de una determinada entidad debe cumplir una serie de reglas para poder ser almacenado en un archivo



Archivo de clientes

- ✓ Registros deben tener longitud fija.
- ✓ Registros deben ser identificados por un valor único e irrepetible.
- ✓ El archivo, en consecuencia al ítem anterior, no debiera admitir registros duplicados.

Archivos

En C cada archivo es visto como con un flujo secuencial de bytes. Al establecer la conexión con el archivo físico en la apertura lo asocia con un flujo, o corriente, que puede ser visto como un dispositivo lógico: no será necesario hacer nuevamente referencia al archivo físico, sino que todas las acciones se realizarán por medio de este flujo. Los flujos proporcionan un canal de comunicación entre los archivos y los programas.

Para las operaciones básicas sobre archivos utilizaremos la biblioteca **<stdio>** (stdio.h)

Archivos: OPERACIONES BASICAS

Archivos - Apertura

Para comenzar a trabajar con el archivo hay que establecer la conexión entre el programa y el archivo, o sea realizar su apertura. La función encargada es ***fopen()*** (del inglés f -file, archivo-, open-abrir-). Su prototipo es:

FILE *p;

P = fopen("ruta_nombre_archivo", "modo_de_apertura")

La función ***fopen()*** devuelve un puntero FILE en caso de que la operación haya sido exitosa; si la apertura no pudo hacerse devuelve la constante ***NULL***.

Archivos - Modos de apertura

El ***modo de apertura*** determina lo que se puede hacer con el archivo. Los modos posibles son:

- “r”: Apertura para lectura de un archivo de texto (read). El archivo debe existir.
- “w”: Apertura para la escritura de un archivo de texto (write). Si el archivo no existe lo crea; si existe lo reemplaza por un archivo vacío. Todos los registros - si los tuviera- se pierden.
- “a”: Apertura de un archivo de texto para agregar registros (append). Si no existe lo crea.

Archivos – Modos de apertura

- “rb”: Apertura para lectura de un archivo binario (read). El archivo debe existir.
- “wb”: Apertura para la escritura de un archivo binario (write). Si el archivo no existe lo crea; si existe lo reemplaza por un archivo vacío. Todos los registros - si los tuviera- se pierden.
- “ab”: Apertura de un archivo binario para agregar registros (append). Si no existe lo crea.

Archivos – Modos de apertura

Características Importantes

- "r": Solo lectura, el archivo debe existir
- "w": Siempre crea un archivo nuevo (borra el existente si hay uno)
- "a": Preserva el contenido existente, escribe al final del archivo
- Si a cualquiera de los modos se le agrega el signo + (por ejemplo "rb+"), se le agrega la funcionalidad que el modo no tiene (con "rb+" se abre un archivo binario para lectura y escritura).
- En sistemas Unix/Linux, la distinción binario/texto no es necesaria
- Siempre se debe verificar que **fopen()** no retorne NULL
- Siempre se debe cerrar los archivos con **fclose()** para liberar recursos

Archivos – Modos de apertura

Modo	Descripción	Archivo Existe	Archivo No Existe
"r"	Lectura (read)	Abre para lectura	Error (NULL)
"w"	Escritura (write)	Sobrescribe contenido	Crea nuevo archivo
"a"	Anexar (append)	Añade al final	Crea nuevo archivo
"r+"	Lectura/Escritura	Abre para leer y escribir	Error (NULL)
"w+"	Escritura/Lectura	Sobrescribe contenido	Crea nuevo archivo
"a+"	Anexar/Lectura	Lee y añade al final	Crea nuevo archivo

Archivos – Modos de apertura

Modo	Descripción
"rb"	Lectura binaria
"wb"	Escritura binaria
"ab"	Anexar binario
"r+b" / "rb+"	Lectura/Escritura binaria
"w+b" / "wb+"	Escritura/Lectura binaria
"a+b" / "ab+"	Anexar/Lectura binaria

Archivos – Cierre

Para cerrar un archivo, esto es, clausurar la vinculación entre el programa y el archivo, se utiliza la función ***fclose()***. El único parámetro que tiene la función es el puntero FILE.

```
fclose(p);
```

Donde *p* sería el hipotético puntero que se utilizó para la apertura del archivo.

Siempre debe cerrarse un archivo al finalizar el uso de la conexión.

Archivos – Escritura

Escribir en un archivo significa copiar un conjunto de datos almacenados en la memoria en un archivo ubicado en un dispositivo externo.

La función que utilizaremos es *fwrite()*.

Sus parámetros son:

```
fwrite(&variable, tamaño_variable, cantidad_registros, punteroFILE)
```


Archivos - Escritura

El significado de cada parámetro es:

- ***&variable***: dirección de la variable que contiene la información que se quiere escribir en el archivo.
- ***tamaño_variable***: tamaño (en bytes) de la variable que contiene la información.
- ***cantidad_registros***: cantidad de registros que se quieren escribir.
- ***punteroFILE***: puntero sobre el que se abrió el archivo en el que se quiere escribir.

Archivos – Lectura

Leer de un archivo significa copiar un conjunto de datos almacenados en un archivo ubicado en un dispositivo externo, y luego escribirlos a la memoria. La función que utilizaremos es *fread()*.

Sus parámetros son:

```
fread(&variable, tamaño_variable, cantidad_registros, punteroFILE)
```

Archivos – Lectura

El significado de cada parámetro es:

- ***&variable***: dirección de la variable donde se desea escribir la información que se leyó del archivo.
- ***tamaño_variable***: tamaño (en bytes) de la variable donde se escribirá la información.
- ***cantidad_registros***: cantidad de registros que se quieren leer del archivo.
- ***punteroFILE***: puntero sobre el que se abrió el archivo que se quiere leer.