Autoestudio: Archivos de Texto

Un archivo es una colección de información que almacenamos en memoria física (disco duro, llave usb, memoria flash, CD). Normalmente usamos los archivos para guardar información a la que queremos acceder en ocasiones pero no siempre. Ejemplos de archivos con los que convives todos los días son archivos de texto ***.txt***, ***.doc***, ***.docx***, ***.rtf*** ; archivos de audio ***.wav***, ***.mp3***, ***.mp4*** ; archivos de video  ***.mov***, **.mpeg** , **.avi**.

En **C** aprenderemos a trabajar con archivos de **texto**.

# Proceso para utilizar archivos

Para poder trabajar con cualquier archivo siempre tenemos que seguir estos 3 pasos, de otra forma no se puede acceder al archivo o puede llegar a corromperse.

1. Abrir Archivo
2. Si el archivo existe, procesarlo:
   1. Escribir
   2. Leer
   3. Modificar
3. Cerrar el archivo

Para cada uno de estos 3 pasos hay funciones específicas que realizan esta labor:

## Abrir archivo

Para abrir un archivo tenemos que hacer 2 cosas, primero decirle al programa que vamos a trabajar con un archivo, esto se hace creando un tipo de dato FILE \* y nombrando la variable con la cual nos referiremos al archivo:

**FILE \*arch;**

Segundo tenemos que abrir el archivo con la función ***fopen*** para decirle a la computadora donde está el archivo que queremos abrir y de qué manera pensamos trabajar con él.

**arch=fopen (path, modo);**

Esta función abre un archivo especificado por **path**. El argumento **modo** especifica cómo es abierto el archivo. Donde **arch**, es el nombre lógico del archivo.

#### Modos para abrir un archivo:

“r” abrir archivo para solo lectura, no puede escribir.

“w” abrir archivo para escritura, sobre escribe todo lo anterior que había en el archivo, no puede leer.

“a” abrir archivo para escritura, agrega lo que se desea insertar al final del archivo.

Entonces un ejemplo de cómo abrir un archivo para **leer** en él sería:

**FILE \*lee;**

**lee=fopen (“C:\Users\Yo\Mis Documentos\spp\hola.txt”, “r”);**

Entonces un ejemplo de cómo abrir un archivo para **escribir** en él sería:

**FILE \*escribe;**

**escribe=fopen (“C:\Users\Yo\Mis Documentos\spp\hola.txt”, “w”);**

Entonces un ejemplo de cómo abrir un archivo para **agregar** en él sería:

**FILE \*agrega;**

**agrega=fopen (“C:\Users\Yo\Mis Documentos\spp\hola.txt”, “a”);**

# Procesando el archivo

## Lectura

Dependiendo de cómo hayamos abierto el archivo **(r,w,a)**, podemos realizar diferentes actividades con él. Si el archivo se abría de modo lectura **“r”** **(read)**, los archivos de texto pueden ser leídos de dos formas:

1. caracter por caracter, usando las funciones:
   1. **char letra = fgetc(arch);**
   2. **fgets(arreglo, n, arch);**

donde letra y arreglo respectivamente son variables del tipo de información con el que queremos trabajar, si estamos leyendo carácter por carácter **letra** será un ***char*** si estamos leyendo un conjunto de caracteres **arreglo** será un arreglo ***char [].***

1. con formato de lectura, usando la función:
   1. **fscanf(arch, formato, argumento);**

La lectura “formateada” de un archivo de texto incluye: enteros, flotantes, cadenas de caracteres y caracteres.

**IMPORTANTE:** cuando se lee de un archivo (“r”) se guarda todo lo que dice el archivo en FILE\* arch y se va sacando por partes. Por ejemplo si en el archivo **corrido.txt** contiene:

*Voy a cantar el corrido   
de un hombre que fue a la guerra   
que anduvo en la sierra herido   
para conquistar su tierra.   
  
Lo conocí en la batalla   
y entre tanta balacera   
el que es revolucionario   
puede morir donde quiera.   
  
El general nos decía:   
"Peleen con mucho valor,   
les vamos a dar parcelas   
cuando haya repartición."*

Y abrimos el archivo con el siguiente código:

**FILE \* arch = fopen(“corrido.txt”, “r”);**

Si el archivo abre de forma exitosa entonces tendremos en arch todo el contenido del archivo, si uso:

**char letra = fgetc(arch);**

**printf(“%c”, letra);**

Entonces la primera letre que se guarda es ‘***V’,*** *si repito el código*

**letra = fgetc(arch);**

**printf(“%c”, letra);**

ahora se despliega la letra ***‘o’*,** que es la letra que sigue en arch ya que***‘V’***ya se sacó,si ahora leo una palabra u oración con:

**char arrgelo[13];**

**fgets(arreglo, 13, arch);**

**puts(arreglo);**

lo que se imprimirá en pantalla es ***‘y a cantar e’*** que son los 13 caracteres que siguen en ***arch*** y estos a su vez también se sacan de ***arch***.

Como se puede ver la lectura de archivos es un recorrido del archivo para el cual repetimos tantas veces como sea necesaria uso de funciones, es por ello que utilizamos ciclos para repetir las instrucciones de lectura hasta que ya no deseemos leer más o hayamos terminado de leer el archivo. Para saber si ya terminamos de leer el archivo utilizamos la función **feof(arch);**  por ejemplo:

**while(!feof(arch)){**

**printf(“%c”, fgetc(arch));**

**}**

Leería todo el archivo letra por letra y lo imprimiría en pantalla, hasta llegar al final del archivo. La función **feof** devuelve 1 si llegó al final del archivo y 0 si no ha llegado.

## Escritura

Si el archivo se abrió a modo de escritura entonces podemos usar las funciones de es escritura

**fputc(letra, arch);**

donde ***letra*** es una variable de tipo char. Esta función escribirá letra en el archivo **arch**

Los datos pueden ser escritos como **cadenas de caracteres** con las funciones ***fputs ()***

**fputs(arreglo, arch);**

donde, **arreglo** es una cadena de caracteres que almacenará una línea del archivo y **arch** es el nombre lógico del archivo.

Los datos pueden ser escritos con **formato** con ***fprintf ()***

**fprintf(arch, formato, argumento);**

donde, **arch** es el nombre lógico del archivo, el **formato** especifica como va ha ser escrita y el **argumento** representa el valor o los valores a escribir.

# Cerrar el archivo

Los archivos que abrimos tanto para escritura o lectura siempre debemos de cerrarlos cuando terminamos de usarlos con la función:

**fclose (arch);**

Si no se cierra correctamente el archivo, se puede corromper el archivo físico además de que se consume la memoría dinámica de la computadora y si se trabaja con archivos grandes esto puede hacer que lo programas fallen.

# Si deseas saber más puedes consultar, donde se explican los temas a mayor detalle:

(Joyanes Aguilar, L. y Zahonero Martínez, I. *Programación en C, C++, Java y UML*. México : McGraw Hill, 2010.) las siguientes secciones:

* Introducción. (pp. 409)
* Flujos. (pp. 410)
* Apuntador FILE. (pp. 410)
* Apertura de un archivo. (pp. 411)
* Funciones de entrada/salida para archivos. (pp. 415)