

Periodo Mayo 2021

Archivos en c

- Buenos días, como están, en este documento vamos a ver como manejar archivos en c++, haciendo asi perdurable los datos e información generados por nuestros programas .

Lo vamos a hacer en dos instancias, primeramente utilizando las funciones nativas heredadas de c y posteriormente con las funciones propias incorporadas a c++

Comencemos...

Tenemos que definir el nombre del archivo, como una variable de tipo puntero, un tipo especial de variable,

Esto se realiza con el identificador FILE

- **FILE *nombre de archivo**

Las primeras funciones están en stdio.h, encontraremos

```
fopen (const char *filename, const char *opentype);
```

la cual se utiliza para la apertura de un fichero en disco

sus parámetros son

- filename, es el nombre del fichero
 - opentype, es la forma en la que se abre dicho archivo
-
- "r" : abrir un archivo para lectura, el fichero debe existir.
 - "w" : abrir un archivo para escritura, se crea si no existe o se sobrescribe si existe.
 - "a" : abrir un archivo para escritura al final del contenido, si no existe se crea.
 - "r+" : abrir un archivo para lectura y escritura, el fichero debe existir.
 - "w+" : crear un archivo para lectura y escritura, se crea si no existe o se sobrescribe si existe.
 - "r+b ó rb+" : Abre un archivo en modo binario para actualización (lectura y escritura).
 - "rb" : Abre un archivo en modo binario para lectura.

La función que se utiliza para cerrar un archivos es

- **fclose (nombre del archivo)**

donde nombre del archivo, es aquel archivo que deseo cerrar

❖ Lectura de un archivo secuencial de texto

una vez que sabemos como abrir y cerrar un archivo secuencial de texto, vamos a centrarnos en como se recupera la información de su contenido, es decir su lectura,

estos se realiza con la **función fgetc()**, la cual **extrae de a uno los caracteres escritos en el archivo abierto anteriormente**

- Esta función lee un caracter a la vez del archivo que esta siendo señalado con el puntero ***archivo**. En caso de que la lectura sea exitosa devuelve el caracter leído y en caso de que no lo sea o de encontrar el final del archivo devuelve **EOF**.

```
fgetc(nombre de archivo);
```

para la correcta lectura de un archivo secuencial tenemos que tener en cuenta que el ultimo carácter colocado por la función fclose() es una marca de fin de archivo EOF end of file, por lo tanto realizaremos la lectura de un archivo MIENTRAS no aparezca la el identificador de EOF.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    FILE *archivo;
    char caracter;

    archivo = fopen("prueba.txt", "r");

    while((caracter = fgetc(archivo)) != EOF)

        printf("%c", caracter);

    fclose(archivo);
}
```

Explicacion

- Se define una variable de tipo puntero denominada archivo
- Se da apertura al archivo con fopen(), de un archivo llamado prueba.txt en modo lectura "r"
- Se utiliza la función fgetc() para extraer el contenido del mismo que será guardado en la variable caracter
- Se realiza esta operación MIENTRAS no se detecte la marca de EOF
- Se muestra el contenido de la variable carácter

- **Escritura de un archivo secuencial de texto**

En general se utiliza la función `fputs()` para escribir texto dentro de este tipo de archivos, ya que permite escribir una cadena completa de texto, facilitando su escritura

```
#include <stdio.h>

int main ()
{
    FILE *fp;

    char cadena[] = "Mostrando el uso de fputs en un
    fichero.\n";

    fp = fopen ( "fichero.txt", "r+" );

    fputs( cadena, fp );

    fclose ( fp );

}
```

Explicación

- Definimos una variable de puntero con FILE denominada fp
- Abrimos el archivo con `fopen()`
- Definimos un vector char donde colocamos el texto a guardar en el archivo
- Escribimos el contenido con `fputs()`
- Cerramos el archivo