**Trabajo de investigación de archivos**

Buscar, definir y ejemplificar los siguientes conceptos adjuntando las imágenes correspondientes

**a) Crear un archivo en C**

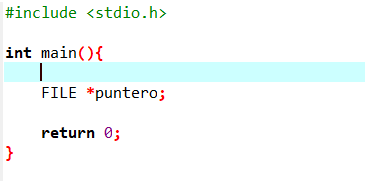
El estándar de C contiene varias funciones para la edición de ficheros, éstas están definidas en la cabecera *stdio.h* y por lo general empiezan con la letra f, haciendo referencia a file. Adicionalmente se agrega un tipo **FILE**, el cual se usará como *apuntador a la información del fichero*. La secuencia que usaremos para realizar operaciones será la siguiente:

1-Crear un apuntador del tipo **FILE \***

2-Abrir el archivo utilizando la función **fopen** y asignándole el resultado de la llamada a nuestro apuntador.

3-Hacer las diversas operaciones (lectura, escritura, etc).

4-Cerrar el archivo utilizando la función **fclose**.



**b) Abrir y cerrar un archivo**

FOPEN()

Antes de abrir un fichero necesitamos declarar un puntero de tipo FILE, con el que trabajaremos durante todo el proceso. Para abrir el fichero utilizaremos la función fopen( ).

Su sintaxis es:

FILE \*puntero;

puntero = fopen ( nombre del fichero, "modo de apertura" );

Donde puntero es la variable de tipo FILE, nombre del fichero es el nombre que daremos al fichero que queremos crear o abrir. Este nombre debe ir encerrado entre comillas. También podemos especificar la ruta donde se encuentra o utilizar un array que contenga el nombre del archivo ( en este caso no se pondrán las comillas ). Algunos ejemplos:

puntero=fopen("DATOS.DAT","r");

puntero=fopen("C:\\TXT\\SALUDO.TXT","w");

Una vez que hemos acabado nuestro trabajo con un fichero es recomendable cerrarlo. Los ficheros se cierran al finalizar el programa pero el número de estos que pueden estar abiertos es limitado. Para cerrar los ficheros utilizaremos la función fclose( );

Esta función cierra el fichero, cuyo puntero le indicamos como parámetro. Si el fichero se cierra con éxito devuelve 0

fclose(puntero);

Un ejemplo ilustrativo aunque de poca utilidad:

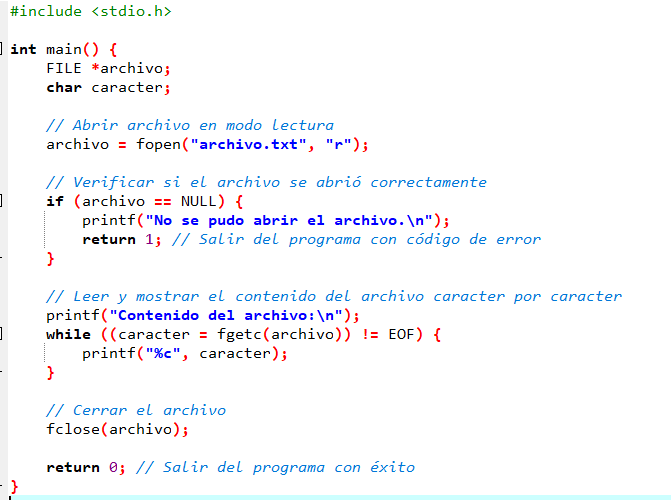
FILE \*pf;

pf=fopen("AGENDA.DAT","rb");

if ( pf == NULL ) printf ("Error al abrir el fichero");

else fclose(pf);

<https://disenowebakus.net/ficheros.php>



**c) Leer, escribir, sobre escribir, etc. (parámetros)**

Una lista de parámetros opentype para la función fopen son:

"r" : abrir un archivo para lectura, el fichero debe existir.

"w" : abrir un archivo para escritura, se crea si no existe o se sobreescribe 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.

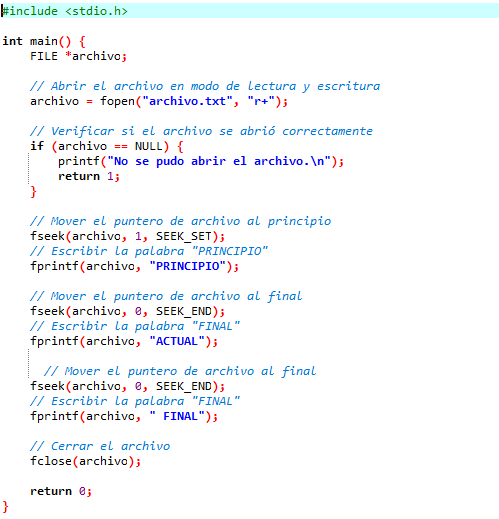
"w+" : crear un archivo para lectura y escritura, se crea si no existe o se sobreescribe si existe.

"r+b ó rb+" : Abre un archivo en modo binario para actualización (lectura y escritura).

"rb" : Abre un archivo en modo binario para lectura.

<https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C/Manejo_de_archivos>

**d) Moverse dentro del fichero y escribir la palabra “PRINCIPIO” , “ACTUAL” O “FINAL” según corresponda.**



**e) Diferencia entre fputc y fputs**

fputc() escribe el carácter c, modelado a un unsigned char, en flujo.

fputs() escribe la cadena de caracteres s en flujo, sin su terminador '\0'.

**Práctica**

1) Realizar un programa en el cual se ingresen los siguientes datos

Nombre, Apellido y DNI

a) Crear un menu donde se ingresen los datos y los guarde en un

archivo

b) Otro ítem donde se pueda realizar una búsqueda por DNI y

devuelva los datos de Nombre y apellido

c) Salir del mismo

2) Modificar el programa anterior de manera que se puedan realizar

búsquedas por Apellido y por nombre a la vez.

3) Agregarle validación de datos de manera que no se repitan cuando sea necesario.

4) Modificar dicho programa de manera que me muestre los datos de manera ordenada Según se desee por nombre y apellido o por DNI.

5) Guardar Los cambios en un archivo con nombre pertinente sin modificar el originaly pudiendo tener acceso a los mismos

6) Realizar un programa que realice las siguientes acciones

a. Guarde el registro del nombre y el puntaje de un jugador (solicitando el ingreso de los datos del usuario)

b. Muestre el resultado de los 10 mejores jugadores ordenados por mayor puntaje.

c. Realice y muestre una búsqueda de un Nombre y devuelva sus resultados

NOTA: el registro debe permanecer aunque se apague la PC.