**Archivos Binarios**

Diagrama

Descripción generada automáticamenteEscritura

**Sintaxis de la función**

**Int fwrite (void \* BUF, size\_t TAM, size\_t NUM, FILE \* FP);**

Siendo:

* BUF la dirección de inicio del vector de lo que sea (Estructuras, enteros …)
* TAM es el tamaño de los bloques que tendría la estructura o el entero.
* NUM es la cantidad de bloques que voy a transferir de una. **99% de los casos es 1.**
* FP es el nombre del puntero asociado al archivo a donde mando los bytes.

“A partir de la dirección de memoria BUF manda al archivo asociado al puntero FP una cantidad NUM de bloques, de tamaño TAM”.

Entonces lo que transfiere a partir de BUF una cantidad de bytes NUM x TAM.

Diagrama

Descripción generada automáticamenteEl tipo int es la cantidad de bloques que se transfirieron en realidad, es decir que debería de coincidir con NUM.

Diagrama

Descripción generada automáticamenteLectura

**Sintaxis de la función**

**Int fread (void \* BUF, size\_t TAM, size\_t NUM, FILE \* FP);**

Estructura de Código ----- “Leo chequeo opero y leo”

Altas

Diagrama

Descripción generada automáticamente con confianza mediaEn el modo de apertura Append, si el archivo no existe lo crea como en escritura, pero si existe, lo abre y ubica la ventana luego del ultimo dato. No me destruye el archivo y me prepara para escritura.

Acceso Random

fseek (FILE \* FP, long DESPLAZAMIENTO, int REFERENCIA);

¡¡¡¡¡¡ El DESPLAZAMIENTO de esta función SIEMPRE está en BYTES!!!!!!!

Cuando se transfiere a través de la ventana, esta siempre naturalmente se mueve hacia el próximo registro.

**Referencias:**

* 0 ORIGEN seek\_set Desde el principio del archivo.
* 1 POSICION ACTUAL seek\_cur En la posición actual donde estoy.
* 2 POSICION FINAL seek\_end En el final del archivo, por fuera de este. Si quiero editar el ultimo registro voy a tener que dar un paso hacia atrás.

Cuando la referencia es el origen, el desplazamiento siempre será positivo.

Cuando la referencia es la posición actual, el desplazamiento será positivo o negativo, según necesidad.

Cuando la referencia es el final, el desplazamiento siempre será negativo.

Modificaciones en el mismo archivo

Describiré a continuación, con mis palabras, la manera en la que va a funcionar el proceso de realizar modificaciones en un mismo archivo.

Vamos a ir recorriendo el archivo con lecturas. Es menester recordar que una vez que se realiza una operación, sea de lectura o de escritura, la ventana va a moverse naturalmente hacia el final del registro el tamaño de la estructura que manejemos.

Por lo tanto, a medida que recorramos vamos a ir encontrando, mediante una condición, los valores que queremos modificar y cuando esto ocurra, los vamos a guardar en el buffer, ya editado según lo que queríamos hacer.

Ahora lo que tenemos que hacer es reescribir ese dato modificado, en su lugar original, pero tenemos que tener en cuenta el movimiento natural de la ventana, la cual se encuentra ya en el lugar del próximo registro. Es por esto que tenemos que volver -1 para atrás para posicionarnos en el registro a editar.