Entradas y Salidas de Datos

Entrada y salida

Además, los archivos binarios permiten el posicionamiento, tanto del puntero de lectura (el lugar de dónde se va a leer) como el de escritura (el lugar del archivo donde se va a escribir). Estos punteros sólo entienden de una posición en bytes en el archivo, y no de registros.

Entrada y salida

Siendo prácticos, para avanzar al registro n, será necesario emplear la fórmula: (n-1) * sizeof(Persona). Así, para el tercer registro:

```
f.seekp( 2 * sizeof( Persona ), ios::beg ); // escritura

f.seekg( 2 * sizeof( Persona ), ios::beg ); // lectura
```

Entrada y salida

Es posible saber en qué punto del archivo se encuentra uno de los punteros mediante los métodos:

```
f.tellg();

f.tellp();
```

Empleo: https://replit.com/join/wmqqwscfij-maximo-arielari

Abrir un archivo

La primera operación que generalmente se realiza sobre un objeto de una de estas clases es asociarlo a un archivo real. Este procedimiento se conoce como abrir un archivo . Un archivo abierto está representado dentro de un programa por un flujo (es decir, un objeto de una de estas clases; en el ejemplo anterior, esto era myfile) y cualquier operación de entrada o salida realizada en este objeto de flujo se aplicará al archivo físico asociado a eso.

Abrir un archivo

Para abrir un archivo con un objeto de flujo, usamos su función miembro open: Where es una cadena que representa el nombre del archivo que se abrirá, y es un parámetro opcional con una combinación de las siguientes banderas:

open (filename, mode);

Abrir un archivo

- **ios::in** Abierto para operaciones de entrada.
- **ios::out** Abierto para operaciones de salida.
- **ios::binary** Abrir en modo binario.
- **ios::ate** Establezca la posición inicial al final del archivo.Si no se establece esta bandera, la posición inicial es el comienzo del archivo.
- **ios::app** Todas las operaciones de salida se realizan al final del archivo, agregando el contenido al contenido actual del archivo.
- **ios::trunc** Si el archivo se abre para operaciones de salida y ya existía, su contenido anterior se elimina y se reemplaza por el nuevo.

Abrir un archivo

Todas estas banderas se pueden combinar usando el operador bit a bit OR (|). Por ejemplo, si queremos abrir el archivo example.binen modo binario para agregar datos, podríamos hacerlo mediante la siguiente llamada a la función miembro open:

```
ofstream myfile;

myfile.open ("example.bin", ios::out | ios::app | ios::binary);
```

archivos binarios

Para archivos binarios, leer y escribir datos con los operadores de extracción e inserción (<<y >>) y funciones como getlineno es eficiente, ya que no necesitamos formatear ningún dato y es probable que los datos no estén formateados en líneas.

Los flujos de archivos incluyen dos funciones miembro diseñadas específicamente para leer y escribir datos binarios secuencialmente: write y read. La primera (write) es una función miembro de ostream(heredada por ofstream). Y read es una función miembro de istream(heredada por ifstream). Los objetos de clase fstream tienen ambos. Sus prototipos son:

write (memory_block, size);

read (memoria_bloque, tamaño);