Archivos y su procesamiento ... Continuación

Archivos en formato binario

Ya sabemos que en una computadora también contamos con archivos cuyo contenido no son carácteres ASCII puros, sino que constan de secuencias de bytes que deben ser interpretados por aplicaciones específicas. Para entenderlo vamos a darnos a continuación un ejemplo bastante simple.

Supongamos que queremos almacenar cantidades enteras en un archivo. El archivo lo vamos alimentar con cierta periodicidad y, llegado el momento, vamos a leer todos esos valores y procesarlos. Podemos elaborar un programa que nos ayude a construir ese archivo de dos maneras:

a) Como un archivo de texto, en el cual guardaremos cada entero separado por un espacio del siguiente y del anterior, o un entero por cada línea del archivo.

b) Como un archivo en el que almacenaremos cada entero, uno a continuación del otro, con el formato binario de cuatro bytes que nos proporciona este lenguaje.

Como podrán darse cuenta, el archivo que se describe en el literal a, es legible desde cualquier programa de visualización o edición de texto sencillo, como Bloc de Notas, Notepad++, Gedit, etc. Un ejemplo de este archivo en formato texto sería:

misEnteros_a.txt:
5
12
24
16
30
8

En el que cada número se ha colocado en una línea diferente del archivo.

Si esa misma secuencia de números se grabara en un archivo con formato binario, no podríamos ver su contenido aunque lo cargáramos con un editor de texto. Simplemente veríamos una serie de símbolos extraños y no interpretables a simple vista.

Como se dijo en el documento anterior, para grabar datos en archivos binarios utilizaremos el método write( ), en cuyo primer argumento debemos hacer una conversión forzada de tipo a puntero a carácter, dado que es un comando para escritura de bloque de carácteres.

Por ejemplo:

Elaborar un programa que almacene enteros en un archivo, de dos maneras:

a) En formato texto.

b) En formato binario.

En cada caso, luego de correr el programa abra el archivo con un editor de textos, como Bloc de Notas paravisualizar su contenido.

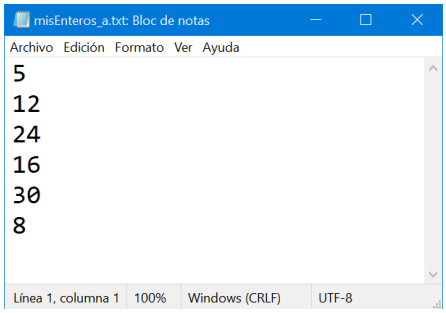
A continuación vemos el programa que crea el archivo de texto de números, un valor por fila:

```
#include <iostream>
#include <fstream>

using namespace std;

int main() {
	int entero;

	cout << endl;
	cout << "GRABAR ENTEROS EN UN ARCHIVO DE TEXTO" << endl << endl;

	ofstream archivo;
	archivo.open("misEnteros_a.txt", ios::app);

	cout << "Digite un entero o Ctrl-Z para finalizar: ";
	while(cin >> entero){
		archivo << entero << '\n';
		cout << "Digite un entero o Ctrl-Z para finalizar: ";
	}

	archivo.close( );

	cout << endl;
	return 0;
}
```

Al ejecutar este programa, e introducir la secuencia de valores que a continuación se ven, tenemos:

```
GRABAR ENTEROS EN UN ARCHIVO DE TEXTO

Digite un entero o Ctrl-Z para finalizar: 5
Digite un entero o Ctrl-Z para finalizar: 12
Digite un entero o Ctrl-Z para finalizar: 24
Digite un entero o Ctrl-Z para finalizar: 16
Digite un entero o Ctrl-Z para finalizar: 30
Digite un entero o Ctrl-Z para finalizar: 8
Digite un entero o Ctrl-Z para finalizar: ^Z
```

El archivo creado, visto en Bloc de Notas, es el siguiente:



A continuación vemos el programa que crea el archivo de números en formato binario:

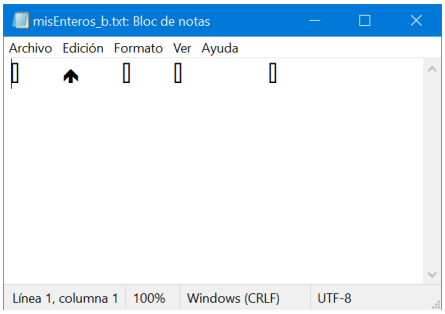
```
#include <iostream>
#include <fstream>

using namespace std;
```

```
int main() {
	int entero;

	cout << endl;
	cout << "GRABAR ENTEROS EN UN ARCHIVO BINARIO" << endl << endl;

	ofstream archivo;
	archivo.open("misEnteros_b.txt", ios::binary | ios::app);

	cout << "Digite un entero o Ctrl-Z para finalizar: ";
	while(cin >> entero){
		archivo.write((char *) &entero, sizeof(int));
		cout << "Digite un entero o Ctrl-Z para finalizar: ";
	}

	archivo.close( );

	cout << endl;
	return 0;
}
```

Al ejecutar este programa, tenemos:

```
GRABAR ENTEROS EN UN ARCHIVO BINARIO

Digite un entero o Ctrl-Z para finalizar: 5
Digite un entero o Ctrl-Z para finalizar: 12
Digite un entero o Ctrl-Z para finalizar: 24
Digite un entero o Ctrl-Z para finalizar: 16
Digite un entero o Ctrl-Z para finalizar: 30
Digite un entero o Ctrl-Z para finalizar: 8
Digite un entero o Ctrl-Z para finalizar: ^Z
```

El archivo creado, visto en Bloc de Notas, es el siguiente:



Note que se le colocó la extensión .txt solo para que al darle doble clic se abriera su contenido con Bloc de Notas. Pero al observar el área de edición del Bloc de Notas nos damos cuenta que no el legibel en lo absoluto. Solo podrá ser leído y procesado por un programa que reconozca en qué formato está su contenido.

Escribamos ahora un programa que lea el archivo misNumeros_b.txt, muestre su contenido y calcule la media aritmética de sus datos. Para leer de un archivo binario utilizaremos el método read( ), cuyos argumentos son los mismos que el método write( ).

```
#include <iostream>
#include <fstream>

using namespace std;
```

```
int main() {
	int entero, suma, n;
	float promedio;

	cout << endl;
	cout << "LEER ENTEROS DE UN ARCHIVO BINARIO" << endl << endl;

	ifstream archivo;
	archivo.open("C:\\Users\\UCA\\Documents\\ProyectosVSCode\\archivo12\\misEnteros_b.txt");

	suma = n = 0;
	cout << "Los datos del archivo binario son:" << endl;
	while(archivo.read((char *) &entero, sizeof(int))){
		cout << entero << endl;
		suma = suma + entero;
		n = n + 1;
	}
	promedio = (float) suma / n;

	cout << "El promedio de los valores es: " << promedio << endl;

	archivo.close( );

	cout << endl;
	return 0;
}
```

Al ejecutar este programa, tenemos:

```
LEER ENTEROS DE UN ARCHIVO BINARIO

Los datos del archivo binario son:
5
12
24
16
30
8
El promedio de los valores es: 15.8333
```

Retomaremos el ejercicio de las notas de estudiantes que vimos hace algunas clases. En esta nueva versión se abrirá un archivo en modo de escritura, el contenido del arreglo de estudiantes se grabará en él y se cerrará el archivo. Luego el mismo archivo se abrirá con otro nombre de flujo en modo de lectura y se recorrerá leyendo sus registros para desplegar la información que cada registro contiene.

De nuevo recordemos que la extensión del archivo físico no interesa, pero que se grabará como .txt solo para facilitar su apertura con Bloc de Notas.

```
#include<iostream>
#include<string>
#include <string.h>
#include <fstream>

const int longCad = 13;

using namespace std;
```

```
struct estudiante{
   char nombre[longCad];
   float nota1;
   float nota2;
   float nota3;
   float notaPromedio;
};

int main(void)
{
   ofstream archivoSalida;
   archivoSalida.open("cuadroNotas.txt", ios::out | ios::binary);

   estudiante registro;
   string cadAux;
   int n, i;

   cout << endl;
   cout << "GRABACION DE ESTRUCTURAS EN UN ARCHIVO" << endl << endl;

   cout << "Cuantos estudiantes? ";
   cin >> n;
   cin.ignore(100, '\n');
   estudiante listado[n];

   cout << "Digite nombre, apellido y las tres notas:" << endl;
   for(i = 0; i < n; i++){
      cout << "Nombres: ";
      getline(cin, cadAux, '\n');
      strncpy(listado[i].nombre, cadAux.c_str( ), longCad-1);
      listado[i].nombre[longCad-1]='\0';
      cout << "Nota 1: ";
      cin >> listado[i].nota1;
      cout << "Nota 2: ";
      cin >> listado[i].nota2;
      cout << "Nota 3: ";
      cin >> listado[i].nota3;
      cin.ignore(100, '\n');
      archivoSalida.write((char *) &listado[i], sizeof(listado[i]));
   }

   archivoSalida.close();

   cout << "Los datos del arreglo son:" << endl;
   for(i = 0; i < n; i++){
      cout << listado[i].nombre << "-->" << strlen(listado[i].nombre) << endl;
      cout << "Nota 1: " << listado[i].nota1 << endl;
      cout << "Nota 2: " << listado[i].nota2 << endl;
      cout << "Nota 3: " << listado[i].nota3 << endl;
   }

   ifstream archivoEntrada;
   archivoEntrada.open("cuadroNotas.txt", ios::in | ios::binary);
   cout << "Contenido del archivo:" << endl;
   while(archivoEntrada.read((char *) &registro, sizeof(listado[i]))){
      cout << registro.nombre << endl;
      cout << registro.nota1 << endl;
      cout << registro.nota2 << endl;
      cout << registro.nota3 << endl;
   }

   archivoEntrada.close();

   cout << endl;
   return 0;
```

}

Una corrida del programa es la siguiente:

GRABACIÓN DE ESTRUCTURAS EN UN ARCHIVO

¿Cuantos estudiantes? 2
Digite nombre, apellido y las tres notas:
Nombres: Ana Tomia
Nota 1: 1
Nota 2: 2
Nota 3: 3
Nombres: Perico Palotes
Nota 1: 4
Nota 2: 5
Nota 3: 6
Los datos del arreglo son:
Ana Tomia-->9
Nota 1: 1
Nota 2: 2
Nota 3: 3
Perico Palot-->12
Nota 1: 4
Nota 2: 5
Nota 3: 6
Contenido del archivo:
Ana Tomia
1
2
3
Perico Palot
4
5
6

Luego de grabar el archivo, al abrirlo con Bloc de Notas se ve lo siguiente: