**Archivos binarios**

En los archivos binarios, los datos se leen o escriben sin ningún tipo de transformación. Es una transferencia directa de memoria a archivo o de archivo a memoria. En C++ la lectura en binario se realiza mediante el método read y la escritura mediante el método write. En ambos casos hay que pasar como primer parámetro un puntero de tipo char a la variable a leer o escribir, y como segundo dato el número de bytes a leer o escribir. La característica de este tipo de archivos es que la longitud de su registro es fija, por lo que están orientados al tipo de dato struct.

FUNCIONES DE I/O EN ARCHIVOS BINARIOS

Función read()

Empleada en archivos de acceso secuencial, permite leer un bloque de bytes desde un archivo de entrada y colocarlos en una dirección específica de memoria.

**Forma de uso**

**typedef struct persona{**

**string nombre;**

**int xclave;**

**float sueldo;**

**}Empleado;**

**ifstream X1;**

**X1.read((char \*)(&X1), sizeof(Empleado));**

**&X1: cast de tipo apuntador char de la dirección en disco del bloque**

**de bytes que se ha de leer para bajar a memoria**

**sizeof(): tamaño del bloque en bytes que se leerán desde el archivo**

Función write()

Empleada en archivos de acceso secuencial, permite escribir un bloque de bytes desde una dirección específica en memoria a un archivo de salida.

**Forma de uso**

**typedef struct persona{**

**string nombre;**

**int xclave;**

**float sueldo;**

**}Empleado;**

**ofstream X2;**

**X1.write((char \*)(&X2), sizeof(Empleado));**

**&X2: cast de tipo apuntador char de la dirección en memoria del bloque**

**de bytes que se ha de escribir en el archivo**

**sizeof(): tamaño del bloque en bytes que se leerán desde el archivo**

**EN LA CARPETA DE ARCHIVOS BINARIOS Y DE ACCESO DIRECTO(ALEATORIOS) SE ILUSTRA CON**

**EJEMPLOS COMPLETOS LO ANTRERIORMENTE DESCRITO.**

**ARCHIVOS BINARIOS DE ACCESO DIRCTO(ALEATORIO)**

**SON ARCHIVOS QUE LEEN/ESCRIBEN EN UNA POSICIÓN DETERMINADA DEL ARCHIVO, DEPENDIENDO DE LA POSICION RELATIVA(ABSOLUTA) DEL APUNTADOR DE ARCHIVO.**

**FUNCIONES DE POSICIONAMIENTO EN RCHIVOS DE ACCESO ALEATORIO**

**Función seekg()**

**Función de posición que posiciona el apuntador de archivo sobre una posición específica en un archivo de entrada. En conjunto con la función read(), posiciona el apuntador de archivo sobre una posición específica del mismo.**

**Forma de uso**

**Ifstream X1;**

**Empleado emp;**

**X1.seekg(sizeof(Empleado), ios::beg); //posiciona el apuntador de archivo al inicio del mismo**

**Función seekp()**

**Función de posición que posiciona el apuntador de archivo sobre una posición específica en un archivo de salida. En conjunto con la función write(), posiciona el apuntador de archivo sobre una posición específica del mismo.**

**Forma de uso**

**ofstream X2;**

**Empleado emp;**

**X2.seekp(sizeof(Empleado), ios::cur); //posiciona el apuntador de archivo en la posición actual**

**del mismo**

**EN LA CARPETA DE ARCHIVOS BINARIOS Y DE ACCESO DIRECTO(ALEATORIOS) SE ILUSTRA CON**

**UN EJEMPLO COMPLETO LO ARRIBA DESCRITO.**

**\*\*\*\*\* OTROS ARCHIBVOS BINARIOS**

Ejemplo: Escritura de 10 enteros en binario.

#include<fstream.h>

#include<iostream.h>

int main()

{

ofstream f;

int i;

f.open("datos.bin");

if(!f)

cout << "Error abriendo el fichero" << endl;

else

{

for(i = 1; i <= 10; i++)

f.write((char \*)(& i), sizeof(i) );

f.close();

}

return 0;

}

Ejemplo completo de programa que crea un archivo binario empleando una estructura y lee el registro escrito en el archivo.

#include<fstream.h>

#include<string.h>

#include<stdio.h>

#include<conio.h>

struct persona{

char nombre[30];

int edad;

float altura;

};

int main()

{

struct persona xval, xva;

ofstream fsalida;

ifstream fentrada;

fsalida.open("prueba.txt",ios::in | ios::out | ios::binary);

fentrada.open("prueba.txt", ios::in | ios::binary);

strcpy(xval.nombre,"nombrex");

xval.edad=40;

xval.altura=1.75;

fsalida.write((char\*)(&xval), sizeof(struct persona));

fentrada.read((char\*)(&xval), sizeof(struct persona));

cout<<" NOMBRE: "<<xval.nombre<<endl;

cout<<" EDAD: "<<xval.edad<<endl;

cout<<"ESTATURA: "<<xval.altura<<endl;

fsalida.close();

fentrada.close();

cin.get();

return 0;

}

#include <fstream.h>

#include <iostream.h>

int main(){

float R=1234.86;

int i,N;

fstream fichbin("F.dat",ios::binary | ios::out); // Apertura como salida

fichbin.write(&R,sizeof(float));

for (i=1;i<=20;i++)

fichbin.write(&i,sizeof(int));

fichbin.close();

fichbin.open("F.dat",ios::binary | ios::in); // Apertura como entrada

fichbin.read(&R,sizeof(float));

cout <<endl << "R= " << R << endl;

for (i=1;i<=20;i++){

fichbin.read(&N,sizeof(float));

cout <<endl << i << "= " << N << endl;

}

}

#include <fstream.h> //librería para manejar la lectura/escritura de archivos

#include<stdio.h>

int main (void) {

int outval=3;

ofstream salida("myfile.txt"); //apertura de archivo en modo escritura

salida << outval; //escribimos el valor de outval en el archivo

salida.close(); //cierre de fichero

int inval;

ifstream lectura("myfile.txt"); //declaración de un archivo en modo lectura

lectura >> inval; //leemos el archivo y su contenido lo colocamos en inval

lectura.close(); //cierre del archivo

cout << "el valor escrito es = " << inval << endl; //el valor escrito es = 3

getch();

return 1;

}