

UNIVERSIDAD DE GUADALAJARA

CUCEI

DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER-HUMANA

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

PRÁCTICA No. 2

TEMA: Archivos de Acceso Aleatorio (Altas)

EQUIPO No. 4

INTEGRANTES:

Munguía Guízar Marlon Uriel

Padilla Perez Jorge Daray

Pérez Palacios Perla Michelle

Pulido Tobías Rafael Agustín

SEMINARIO DE SOLUCIÓN DE PROBLEMAS DE ESTRUCTURAS DE DATOS II | SECCIÓN D19 | 2022B

PROF. Mariscal Lugo Luis Felipe

MARCO TEÓRICO

**Los sistemas de reservación de aerolíneas, sistemas bancarios, sistemas de punto de venta, cajeros automáticos y demás tipos de sistemas de procesamiento de transacciones son ejemplos de aplicaciones de:**

* Aplicaciones comunes de acceso instantáneo.

**Como lo hemos dicho antes C++ no impone una estructura en un archivo. De esta forma, las aplicaciones que deseen utilizar archivos de acceso aleatorio deben crearlos. Puede utilizarse una variedad de técnicas para crear archivos de este tipo.**

**Tal vez el método más sencillo sea requerir que todos los registros en un archivo sean:**

* Que todos los registros de un archivo sean de la misma longitud.

**Dibujar la Fig. 17.9 de la pág. 722 del libro del maestro Deitel 6ta. Edición**

Diagrama, Gráfico de cajas y bigotes

Descripción generada automáticamente

**¿Para qué sirve la función** **seekp( )?**

* Para ostream seekp (“seek put”, ”buscar colocar”) cada objeto ostream tiene un apuntador put que indica el numero del byte en el que el archivo se debe colocar la siguiente operación de salida.

**¿Para qué sirve la función** **write( )?**

* La función del miembro write de ofstream envía un número fijo de bytes, empezando en una ubicación específica en memoria, hacia el flujo específico. cuando el flujo se asocia con un archivo, La función write Escribe los Datos en la posición dentro del archivo especificado por el apuntador “put” de posición de archivo.

**¿Para qué sirve el operador** **reinterpret\_cast?**

* Para convertir la dirección de un entero en un apuntador const \* char.
* Para realizar conversiones entre tipos que los operadores de conversión originales qué C++  heredó C. Hablamos sobre los espacios de nombres, una característica especialmente imprescindible para los desarrolladores de software que construyen sistemas considerables, en especial aquellos quienes crean sistemas a partir de las bibliotecas de clases.

**Bibliografía.**

Deitel., H. M. & Deitel., P. J. (2008). *C++: cómo programar* (6.a ed.). Pearson.

CÓDIGO FUENTE

Main():

1 /\*Equipo 4:

2 MunguÃ­a GuÃ­zar Marlon Uriel

3 Padilla Perez Jorge Daray

4 PÃ©rez Palacios Perla Michelle

5 Pulido TobÃ­as Rafael AgustÃ­n

6 SecciÃ³n: D19

7 Calendario: 2022B

8 \*/

9 //CÃ³mo escribir en un archivo de acceso aleatorio

10 #include <iostream>

11 #include<iomanip>

12 #include<fstream>

13 #include<cstdlib>

14 #include <Deportista.h>

15

16 **using namespace std**;

17

18 **int** main()

19 {

20 **int** numeroDeSocio;

21 **char** nombre[15];

22 **int** edad;

23 **char** deporte[15];

24

25 Deportista d;//un objeto o registro Deportista

26

27 **ofstream** archDeportSalida("deportistas.dat", ios::binary);//El constructor abre el archivo

28

29 //salir del programa si ofstream no puede abrir el archivo

30 **if**(!archDeportSalida){

31 **cerr**<<"No se pudo abrir el archivo."<<**endl**;

32 exit(1);

33 }//if

34 **cout**<<"Escriba el numero de socio(De 1 a 100, 0 para terminar la entrada)\n?";

35 **cin**>>numeroDeSocio;

36 d.establecerNumeroDeSocio(numeroDeSocio);

37 //El usuario proporciona la informaciÃ³n que se guarda en el archivo

38 **while**(d.obtenerNumeroDeSocio()>=1 && d.obtenerNumeroDeSocio()<=100){

39 **cout**<<"Teclea nombre, edad y deporte\n?";

40 **cin**>>**setw**(15)>>nombre;

41 **cin**>>edad;

42 **cin**>>**setw**(15)>>deporte;

43 //establecer los valores nombre, edad y deporte del registro

44 d.establecerNombre(nombre);

45 d.establecerEdad(edad);

46 d.establecerDeporte(deporte);

47 //buscar la posicion en el archivo de registro especificado por el usuario

48 archDeportSalida.seekp((d.obtenerNumeroDeSocio()-1)\***sizeof**(Deportista)); //seekp(n, ios::beg);

49 //escribir la informaciÃ³n especificada por el usuario en el archivo

50 archDeportSalida.write(**reinterpret\_cast**<**const char** \*>(&d),**sizeof**(Deportista));

51 //Permitir que el usuario especifiqueotro numero de socio

52 **cout**<<"Escriba el numero de socio (De 1 a 100, 0 para terminar la entrada)\n?";

53 **cin**>>numeroDeSocio;

54 d.establecerNumeroDeSocio(numeroDeSocio);

55 }//while

56 **return** 0; //El destructor cierra el archivo

57 }//main

**Deportista.h:**

1 /\*Equipo 4:

2 MunguÃ­a GuÃ­zar Marlon Uriel

3 Padilla Perez Jorge Daray

4 PÃ©rez Palacios Perla Michelle

5 Pulido TobÃ­as Rafael AgustÃ­n

6 SecciÃ³n: D19

7 Calendario: 2022B

8 \*/

9 #ifndef DEPORTISTA\_H

10 #define DEPORTISTA\_H

11 #include <cstring>

12 #include <string>

13

14 **using namespace std**;

15 **using std**::**string**;

16

17

18 **class** Deportista

19 {

20 **public**:

21 //constructor predeterminado

22 Deportista(**int** = 0, **string** = "", **int** = 0, **string** = "");

23

24 //funciones de acceso para NumeroDeSocio

25 **void** establecerNumeroDeSocio(**int**);

26 **int** obtenerNumeroDeSocio() **const**;

27

28 //funciones de acceso para nombre

29 **void** establecerNombre(**string**);

30 **string** obtenerNombre() **const**;

31

32 //funciones de acceso para edad

33 **void** establecerEdad(**int**);

34 **int** obtenerEdad() **const**;

35

36 //funciones de acceso para deporte

37 **void** establecerDeporte(**string**);

38 **string** obtenerDeporte() **const**;

39

40

41 **protected**:

42

43 **private**:

44 **int** numeroDeSocio;

45 **char** nombre[15];

46 **int** edad;

47 **char** deporte[15];

48

49 };//Fin de la clase Deportista

50

51 #endif // DEPORTISTA\_H

**DeportistaCPP:**

1 /\*Equipo 4:

2 MunguÃ­a GuÃ­zar Marlon Uriel

3 Padilla Perez Jorge Daray

4 PÃ©rez Palacios Perla Michelle

5 Pulido TobÃ­as Rafael AgustÃ­n

6 SecciÃ³n: D19

7 Calendario: 2022B

8 \*/

9 #include "Deportista.h"

10 #include <cstring>

11

12 **using namespace std**;

13

14 //Constructor

15 Deportista::Deportista(**int** valorNumeroDeSocio, **string** valorNombre, **int** valorEdad, **string** valorDeporte)

16 {

17 //ctor

18 establecerNumeroDeSocio(valorNumeroDeSocio);

19 establecerNombre(valorNombre);

20 establecerEdad(valorEdad);

21 establecerDeporte(valorDeporte);

22 }//fin constructor

23

24 //Obtiene el valor del nÃºmero de socio

25 **int** Deportista::obtenerNumeroDeSocio() **const**{

26 **return** numeroDeSocio;

27 }//fin int obtenerNumeroDeSocio

28

29 //Establece el valor del nÃºmero de Socio

30 **void** Deportista::establecerNumeroDeSocio(**int** valorNumeroDeSocio){

31 numeroDeSocio=valorNumeroDeSocio;

32 }//fin void establecerNumeroDeSocio

33

34 //Obtiene el valor del nombre

35 **string** Deportista::obtenerNombre() **const**{

36 **return** nombre;

37 }//fin string obtenerNombre

38

39 //Establece el valor del nombre

40 **void** Deportista::establecerNombre(**string** cadenaNombre){

41 **const char**\*valorNombre = cadenaNombre.data();

42 **int** longitud = cadenaNombre.**size**();

43 longitud = (longitud < 15 ? longitud : 14);

44 strncpy(nombre, valorNombre, longitud);

45 nombre[longitud] = '\0';

46 }//fin void establecerNombre

47

48 //Obtiene el valor de la edad

49 **int** Deportista::obtenerEdad() **const**{

50 **return** edad;

51 }//fin int obtenerEdad

52

53 //Establece el valor de la edad

54 **void** Deportista::establecerEdad(**int** valorEdad){

55 edad=valorEdad;

56 }//fin void establecerEdad

57

58

59 //Obtiene el valor del deporte

60 **string** Deportista::obtenerDeporte() **const**{

61 **return** deporte;

62 }//fin string obtenerDeporte

63

64 //Establece el valor del deporte

65 **void** Deportista::establecerDeporte(**string** cadenaDeporte){

66 **const char**\*valorDeporte = cadenaDeporte.data();

67 **int** longitud = cadenaDeporte.**size**();

68 longitud = (longitud < 15 ? longitud : 14);

69 strncpy(deporte, valorDeporte, longitud);

70 deporte[longitud] = '\0';

71 }//fin void establecerDeporte

IMAGENES DEL ARCHIVO FÍSICO:

Tabla

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

IMÁGEN DE LA CORRIDA DEL PROGRAMA:

Texto

Descripción generada automáticamente

**Conclusiones:**

* **Munguía Guízar Marlon Uriel.**
* **Padilla Perez Jorge Daray.**

Con respecto a la actividad, la realización del programa nuevo de acceso aleatorio, se le agregaron nuevos métodos funcionales importantes como el write, reinterpret\_cast y seekp, tanto para escribir en el archivo como para que el puntero pueda ser posicionado a gusto del programador, como buena introducción a posibles programas.

* **Pérez Palacios Perla Michelle.**

En esta práctica sobre archivos pudimos conocer sobre las funciones write y seekp que nos ayuda a poner un puntero en la posición específica del archivo y el operador reinterpret\_cast.

* **Pulido Tobías Rafael Agustín.**