

Algoritmos y Estructuras de Datos

Archivos

Universidad Peruana de Ciencias Aplicadas

2020

Outline

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

1. Logro

2. Archivos

3. Archivos de acceso aleatorio

4. Archivos de acceso secuencial

5. Referencias

Outline I

Logro

Archivos

Archivos de acceso aleatorio

Archivos de acceso secuencial

Referencias

1. Logro
2. Archivos
3. Archivos de acceso aleatorio
4. Archivos de acceso secuencial
5. Referencias

Logro

Archivos

Archivos de acceso aleatorio

Archivos de acceso secuencial

Referencias

Al finalizar la sesión el estudiante implementará aplicaciones con datos persistentes usando archivos de manera eficiente.

Outline I

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

1. Logro
- 2. Archivos**
3. Archivos de acceso aleatorio
4. Archivos de acceso secuencial
5. Referencias

A tomar en cuenta

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

- ▶ Almacenar datos en variable o en arreglos es temporal.
- ▶ Al terminar un programa, los datos almacenados se pierden.
- ▶ Para conservar datos permanentemente se utilizan archivos.
- ▶ Los archivos almacenan en dispositivos de almacenamiento secundario, especialmente en discos.
- ▶ Cuando se trabaja con archivos se utilizan buffers, almacenes temporales de datos en memoria.
- ▶ Las operaciones de lectura se hacen a través del buffer, y solo cuando el buffer se llena se realiza la escritura en el disco y se vacía el buffer.

Beneficios

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

- ▶ Permite alcanzar gran cantidad de información en un medio rápidamente accesible por la computadora.
- ▶ Evita el hecho de tener que volver a leer datos.
- ▶ Los datos almacenados pueden ser compartidos por más de un programa.

Operaciones con archivos

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

- ▶ Para operar con un archivo hay que referirse a él mediante el nombre.
- ▶ Las operaciones básicas para usar un archivo son:
 - ▶ Apertura, dándole nombre y otras especificaciones.
 - ▶ Escritura de datos.
 - ▶ Lectura de datos.
 - ▶ Modificación de datos.
 - ▶ Cierre.

Cómo definir un archivo en C++

Logro

Archivos

Archivos de acceso aleatorio

Archivos de acceso secuencial

Referencias

- ▶ En primer lugar debemos incluir el archivo de cabecera `fstream` y declarar el uso del espacio de nombres `std` para abreviar.
- ▶ Ésta librería incluye 3 clases:
 - ▶ `ofstream`: para operaciones de solo escritura.
 - ▶ `ifstream`: para operaciones de solo lectura.
 - ▶ `fstream`: para operaciones de lectura y escritura.

- ▶ Existen 2 formas de abrir un archivo de cualquiera de los 3 tipos.
- ▶ Cualquiera de las 2 formas requiere 2 parámetros: nombre de archivo y modos de apertura. Los modos de apertura son opcionales.

- ▶ Usando el constructor:

```
fstream archivo("nombre.archivo", ios::binary | ios::in);
```

- ▶ Usando método open:

```
ofstream archivo;  
archivo.open("nombre.archivo", ios::binary);
```

- Para verificar si el archivo fue abierto correctamente, se usa el método `is_open()`

Modos de apertura

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

- ▶ `ios::in`: Operaciones de lectura.
- ▶ `ios::out`: Operaciones de escritura.
- ▶ `ios::binary`: Archivo binario.
- ▶ `ios::ate`: Posiciona el cursor al final del archivo.
- ▶ `ios::app`: Operaciones de escritura se realizan al final del archivo.
- ▶ `ios::trunc`: Remplaza el archivo, sobre-escribe su contenido si existiera.

Cerrar un archivo

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

Para cerrar un archivo se usa el método `close()`.

```
archivo.close();
```

Outline I

Logro

Archivos

**Archivos de
acceso
aleatorio**

Archivos de
acceso
secuencial

Referencias

1. Logro
2. Archivos
- 3. Archivos de acceso aleatorio**
4. Archivos de acceso secuencial
5. Referencias

Archivos binarios

Logro

Archivos

**Archivos de
acceso
aleatorio**

Archivos de
acceso
secuencial

Referencias

- ▶ También conocidos como archivos binarios.
- ▶ Permiten acceder a cualquier parte del fichero en cualquier momento como si fueran vectores en memoria.
- ▶ Las operaciones de lectura y (o) escritura pueden hacerse en cualquier punto del archivo.

Escritura de datos

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

Para escribir se usa el método ostream& write (const char* s, streamsize n)

Parámetros

s es un puntero a un arreglo de por lo menos n bytes.

n es el número de bytes a escribir.

Es importante destacar que se debe hacer casting a char* en caso el dato que se desea escribir sea de otro tipo.

```
long numero;  
int arreglo[TAMANO];  
archivo.write((char*)&numero, sizeof(long));  
archivo.write((char*)arreglo, sizeof(int) * TAMANO);
```

Lectura de datos

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

Para leer se usa el método `istream& read (char* s, streamsize n)`

Parámetros

s es un puntero a un arreglo de por lo menos `n` bytes.

n es el número de bytes a escribir.

Es importante destacar que se debe hacer casting a `char*` en caso el dato que se desea leer sea de otro tipo.

```
long numero;  
int arreglo[TAMANO];  
archivo.read((char*)&numero, sizeof(long));  
archivo.read((char*)arreglo, sizeof(int) * TAMANO);
```


Operaciones complementarias

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

- ▶ `bad()` retorna verdadero en caso de error de lectura y escritura. `fail()` igual que el anterior pero además en caso de error de formato. `eof()` verdadero cuando el cursor de lectura se encuentra al final del archivo. `good()` verdadero cuando todos los anteriores retornan falso.
- ▶ `tellg()` y `tellp()`, retornan la posición del cursor de lectura (`get`) y escritura (`put`), respectivamente.
- ▶ `seekg()` y `seekp()`, permiten ubicar los cursores de lectura y escritura en la posición indicada. Pueden recibir una posición absoluta o relativa. En caso sea relativa se usan las banderas `ios::beg`, `ios::end` y `ios::cur` para indicar el punto de referencia a partir del cual se realizará el desplazamiento.

Consideraciones

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

- ▶ Es posible guardar datos de tipos de datos distintos, como en el ejemplo anterior que se escribe un valor de tipo long y luego un arreglo de TAMANO elementos de tipo entero.
- ▶ En caso se guarden tipos de datos distintos, se recomienda hacerlo en un orden determinado previamente establecido.
- ▶ Se recomienda el uso de registros para simplificar el proceso de lectura y escritura.

Outline I

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

1. Logro
2. Archivos
3. Archivos de acceso aleatorio
- 4. Archivos de acceso secuencial**
5. Referencias

Archivos de texto

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

- ▶ La información sólo puede leerse y escribirse empezando desde el principio del archivo.
- ▶ Son más sencillos de manejar, ya que requieren menos funciones, además son más rápidos.
- ▶ Son útiles cuando sólo se quiere almacenar cierta información a medida que se recibe, y no interesa analizarla en el momento.

Escritura y lectura de texto

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

Just like cout y cin

Para escribir y leer archivos de texto se usa la misma sintaxis usada con los objetos cin y cout.

```
ofstream archivo("archivo.txt");  
archivo << "Hola mundo!" << endl;  
  
ifstream archivo("datos.txt");  
archivo >> numero;
```

Formatos de archivos de texto

Logro

Archivos

Archivos de acceso aleatorio

Archivos de acceso secuencial

Referencias

- ▶ Existen muchos formatos de archivos de texto como texto simple, xml, html, csv, tsv y otros.
- ▶ CSV es un formato muy utilizado y tienen la siguiente forma:

```
cabecera1,cabecera2,cabecera3  
dato 1 1,dato 1 2,dato 1 3  
dato 2 1,dato 2 2,dato 2 3
```

Outline I

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias

1. Logro
2. Archivos
3. Archivos de acceso aleatorio
4. Archivos de acceso secuencial
- 5. Referencias**

Referencias

Logro

Archivos

Archivos de
acceso
aleatorio

Archivos de
acceso
secuencial

Referencias



Thomas H. Cormen, Charles E. Leirserson, Ronald L. Rivest, Clifford Stein.
Introduction to Algorithms. Third edition, The MIT Press, Cambridge,
Massachusetts, 2009.



Bjarne Stroustrup. **Programming: Principles and practice using C++**.
Addison-Wesley, Upper Saddle River, NJ, Boston, 2009. Capítulo 19 sección 3,
p. 656.



cplusplus.com **fstream**. <http://www.cplusplus.com/reference/fstream/>



cplusplus.com **Input/Output with files**.
<http://www.cplusplus.com/doc/tutorial/files/>