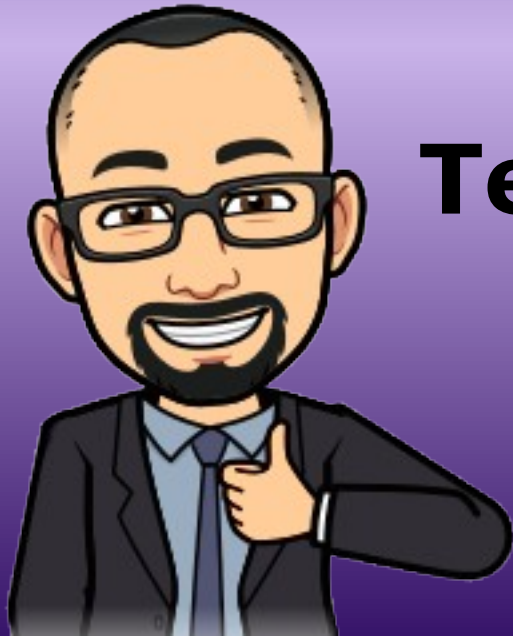


LENGUAJE C++



Tema 5 - Archivos

Archivos

- El **Archivo** es un concepto lógico que es INDEPENDIENTE del dispositivo
 - terminal, disco, cinta, puerto de comm, etc.
- El C++ opera con el paradigma ORWC;
 - *Open*: asigna un buffer y lo asocia a un **Stream**.
 - *Read/Write*: copia de/a los buffers (apuntado por el stream) y mueve la posición de los punteros
 - *Close*: Limpia los buffers, libera el stream.

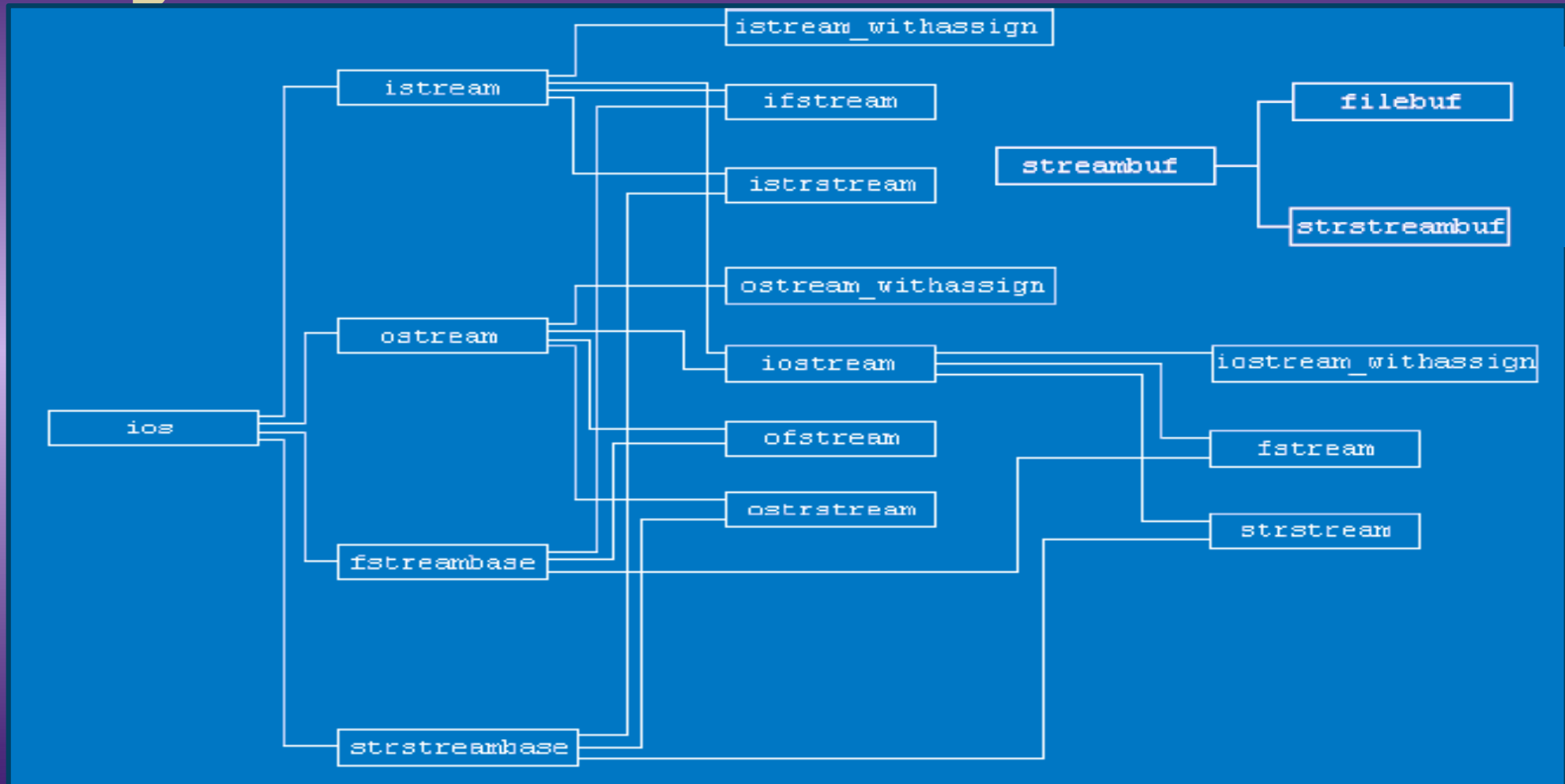
	Texto	Binario
Secuencia de	Caracteres	Bytes
Representación	De la fuente	Del destino
Nro de Bytes leídos de la fuente	Distintos a la cantidad de bytes de la fuente(Traducción)	Igual a la cantidad de bytes de la fuente (No Traduce)

Streams

En C++ el manejo de todas las funciones de E/S (teclado, pantalla y Archivos de disco) las realizan varias clases

- ***streambuf***: manipulación de buffers. Da la interfaz entre la memoria y los dispositivos físicos.
- ***ios***: entradas y salidas. Como clase base, tiene las siguientes derivaciones:
 - ***iostream***: encapsula E/S por teclado y pantalla.
 - ***fstream***: E/S desde Archivos.
 - ***ifstream***: especializada en archivos de entrada.
 - ***ofstream***: especializada en archivo de salida.

Diagrama de Clases ios



Apertura y Cierre de Archivo

- *void open (const char *NombreArchivo, modo);*
 - Modos:
 - Solo Lectura; *ios_base::out*
 - Solo Escritura: *ios_base::app*
 - Lectura y Escritura: *ios_base::in | ios_base::out*
 - Añadir: *ios_base::app*
 - Tipo de Archivo: Por default es TEXTO, para usar Binario se debe agregar al Modo “| *ios::binary*”.
 - Si no puede abrir el archivo, el stream será nulo.
- *void close();*
 - *salida.close();*
- *int eof() const;*
 - Retorna 0 si está en el EOF.
- *void clear(int nState = 0);*
 - Limpia todos los flags de error dentro del stream;

Lectura/Escritura de Archivos de Texto

- Un *Archivo de texto (ArT)* es un Archivo binario con un rango limitado para los valores que puede almacenar.
 - Puede almacenar cualquier dato pero se usa para almacenar datos editables con un editor de texto.
 - La Lectura/Escritura se hace con “<<” y “>>”.
 - La Lectura “>>” realiza conversiones de caracteres:
 - Interpreta los espacios, tabuladores y retornos de línea como separadores, y los elimina de la cadena de entrada.
 - El **problema** surge con el Fin de Archivo que el sistema operativo usa en un ArT. En Microsoft ese valor es **0x1A**.
 - Si en modo Texto leemos un dato interno con ese valor, no se podrá leer mas allá de esa posición.
- Abrir en *modo Binario* evita estos problemas

Lectura/ Escritura de Archivos Binarios

Los Archivos binarios son más útiles para guardar información cuyos valores no estén limitados: imágenes, BD, Estructuras complejas (mix de cadenas y números).

- **Apertura** en Modo Binario: tenemos que añadir el valor `ios::out` e `ios::in`, respectivamente, al valor `ios::binary`.
 - `ofstream fsalida("prueba.dat", ios::out | ios::binary);`
 - `ifstream fentrada("prueba.dat", ios::in | ios::binary);`
- La **Lectura** se realiza con:
 - `istream& get(char& rch);`
 - *Rch*: un puntero a carácter. Allí estará el carácter leído
 - `istream& read(unsigned char* puch, int nCount);`
 - *Puch*: Puntero a una estructura de datos
 - *nCount*: Máxima cantidad de caracteres que se leerán.
- La **Escritura** se realiza con:
 - `ostream& put(char ch);`
 - *Ch*: El carácter a insertar.
 - `ostream & write(const unsigned char* puch, int nCount);`
 - *Puch*: Puntero a una estructura de datos.
 - *nCount*: Máxima cantidad de caracteres que se escribirán.

Ejemplo Modo Texto

```
#include <iostream>
#include <fstream>
#include <stdlib.h>
using namespace std;

int main()
{ char cadena[128];
  ofstream salida("salida.txt"); // Crea un Archivo de salida

  salida << "Hola Matias" << endl;
  salida.close(); // Cerrar el Archivo, para luego poder abrirlo para lectura:

  ifstream Entrada("salida.txt"); // Abre un Archivo de entrada
  Entrada >> cadena; // Leemos mediante el >> que sólo lee parte de la cadena:
  cout << cadena << endl;
  return 0;
}
SALIDA POR CONSOLA "Hola"
```


Ejemplo Modo Binario

```
#include <iostream>
#include <fstream>
#include <string.h>
using namespace std;
struct tipoRegistro {
    char nombre[32];
    int edad;
    float altura;};
int main()
{ tipoRegistro pepe, pepe2;
  ofstream fsalida("prueba.dat", ios::out | ios::binary);
  strcpy(pepe.nombre, "Matias Garcia");
  pepe.edad = 42;
  pepe.altura = (float) 1.78;
  fsalida.write((char *)&pepe, sizeof(tipoRegistro));
  fsalida.close();
  ifstream fentrada("prueba.dat", ios::in | ios::binary);
  fentrada.read((char *)&pepe2, sizeof(tipoRegistro));
  cout << pepe2.nombre << endl;
  cout << pepe2.edad << endl;
  cout << pepe2.altura <<endl;
  fentrada.close();
  return 0;
}
```

SALIDAS
Matias Garcia
42
1.78

Acceso Aleatorio

Acceso aleatorio: hacer lecturas/escrituras en cualquier punto.

- Hemos trabajado con archivos secuenciales: empezamos a leer/escribir desde el principio y avanzamos a medida que leemos/escribimos en ellos hasta el final.
- **streampos tellp()**; informa la posición dentro del fichero de salida.
 - **streamoff** es un **typedef** equivalente a **long**.
- **streampos tellg()**; informa la posición dentro del fichero de entrada.
- **ostream& seekp(streamoff offsett, ios::seek_dir dir);** Cambia la posición del stream de salida.
 - *Offset*: Nueva posición de desplazamiento
 - *Dir*: La dirección de la posición:
 - **ios::beg** Seek desde el Inicio del stream.
 - **ios::cur** Seek desde la posición actual del stream.
 - **ios::end** Seek desde el final del stream.
- **istream& seekg(streamoff off, ios::seek_dir dir);** Cambia la posición del stream de entrada.

Ejemplo Acceso Aleatorio

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{ int i; char cad[20];
  streampos pos; //guardará la posición del archivo
  char mes[][20] = {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio",
  "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"};
  cout << "Crear archivo de nombres de meses:" << endl;
  ofstream fsalida("meses.dat", ios::out | ios::binary); // fichero con los meses:
  for(i = 0; i < 12; i++)
    fsalida.write(mes[i], 20);
  fsalida.close();
  cout << "\nAcceso secuencial:" << endl;
  ifstream fentrada("meses.dat", ios::in | ios::binary);
  fentrada.read(cad, 20);
  do {
    cout << cad << endl;
    fentrada.read(cad, 20);
  } while(!fentrada.eof());
  fentrada.clear(); //limpia los flags errores
```

Ejemplo Acceso Aleatorio

```
cout << "\nAcceso aleatorio:" << endl;
for(i = 11; i >= 0; i--)
{   fentrada.seekg(20*i, ios::beg);
    fentrada.read(cad, 20);
    cout << cad << endl;
}
```

```
// Calcular el número de elementos almacenados en un fichero:
// El número de registros es el tamaño en bytes dividido entre el
// tamaño del registro:
fentrada.seekg(0, ios::end); // ir al final del fichero
pos = fentrada.tellg(); // leer la posición actual
cout << "\nNúmero de registros: " << pos/20 << endl;
fentrada.close();
return 0;
}
```

Ejemplo Flujo hacia el Printer

```
#include <iostream>
#include "stdafx.h"
#include <fstream>
using namespace std;

int main(int argc, char* argv[])
{
    ifstream ArchIN("ReadMe.txt");
    ofstream Printer("LPT1");
    char cad[80];

    cout << "\nComienza Impresion a PRINTER:" << endl;
    ArchIN.read(cad, 80);
    do {
        Printer.write(cad, 80);
        ArchIN.read(cad, 80);
    } while(!ArchIN.eof());
    ArchIN.clear(); //limpia los flags errores
    return 0;
}
```

Bibliografía & Licencia

- ❖ *Como programar en C++, 9na Ed*, Deitel, H.M. y Deitel, P.J., Pearson
- ❖ *Programación en C++, Un enfoque práctico*, Joyanes Aguilar, L., McGraw-Hill.
- ❖ *Thinking in C++, 2da Ed*, Bruce Eckel, Prentice Hall PTR.
- ❖ Este documento se encuentra bajo Licencia Creative Commons Attribution – NonCommercial - ShareAlike 4.0 International (CC BY-NC-SA 4.0), por la cual se permite su exhibición, distribución, copia y posibilita hacer obras derivadas a partir de la misma, siempre y cuando se cite la autoría del **Prof. Matías E. García** y sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.
- ❖ Autor:

Matías E. García

Prof. & Tec. en Informática Aplicada

www.profmatiasgarcia.com.ar

info@profmatiasgarcia.com.ar



www.profmatiasgarcia.com.ar