



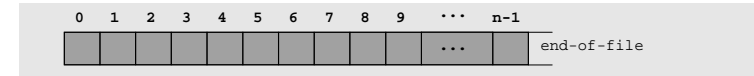
FACULTAD DE INFORMÁTICA

# Ficheros



## Ficheros y flujos de datos (streams)

- C++ considera los ficheros como una secuencia de bits
  - Los ficheros no tienen estructura de registros
  - El fichero acaba cuando se encuentra la marca de fin de fichero (EOF, *end-of-file*)



- Cuando se abre un fichero se crea un objeto stream asociado con el
  - El stream se ocupa de la comunicación entre el programa y el fichero o dispositivo
  - Los flujos estandar como `cin`, `cout`, etc. se crean automáticamente al incluir `<iostream>`



## Ficheros y flujos de datos (streams)

- Los flujos de datos simplifican la operación de entrada salida ya que se ocupan de los detalles de dicha operación
- El trabajo con ficheros es similar al trabajo con la entrada y salida estándar
  - Escribir objetos en un fichero se hace de la misma forma que en pantalla



## Ficheros y flujos de datos (streams)

- Para trabajar con ficheros hay que incluir las bibliotecas `<iostream>` y `<fstream>`
  - Incorporan plantillas generales de clases
    - `basic_ifstream` (entrada o lectura de datos)
    - `basic_ofstream` (salida o escritura de datos)
    - `basic_fstream` (entrada/salida)
  - También incluyen tipos concretos especializados que permiten realizar directamente la entrada/salida de caracteres
    - `ifstream` (entrada de datos en formato `char`)
    - `ofstream` (salida de datos en formato `char`)
    - `fstream` (entrada/salida en formato `char`)



## Ficheros secuenciales de texto: apertura y cierre

- Creación del flujo directamente asociado al fichero  
`ofstream flujoSalida("nombreFichero", modoApertura);`
- Declaración del flujo y asociación posterior al fichero  
`ofstream flujoSalida;`  
`flujoSalida.open("nombreFichero", modoApertura);`
- Cierre de un fichero  
`flujoSalida.close();`
  - El fichero se cierra automáticamente cuando se destruye el flujo asociado
- Modos de apertura

Modo	Descripción
<code>ios::app</code>	Escribir los datos añadiendo al final del fichero.
<code>ios::ate</code>	Abrir el fichero y situarse al final del mismo. No obstante los datos pueden escribirse en cualquier parte del fichero
<code>ios::in</code>	Abrir un fichero para lectura
<code>ios::out</code>	Abrir un fichero para escritura
<code>ios::trunc</code>	Si el fichero existe se destruye su contenido (operación por defecto de <code>ios::out</code> )
<code>ios::binary</code>	Abrir un fichero para entrada/salida binaria

Laboratori

4



## Operadores y ficheros

- El operador `!` está sobrecargado
  - `!flujoSalida`
  - Devuelve un resultado distinto de cero (cierto) si se ha producido algún error (se han activado los indicadores `badbit` o `failbit`)
    - Por ejemplo, cuando se trata de abrir para escritura un archivo que no existe o no se tienen de los permisos para hacerlo
- El operador `void*` está sobrecargado
  - Convierte el flujo en un puntero
  - Devuelve `0` cuando se han activado los indicadores `badbit` o `failbit` y distinto de cero en otro caso
  - Cuando se encuentra la marca EOF se activa el indicador `failbit`
  - `while ( cin >> unaVariable )`
    - El bucle se ejecuta hasta que se encuentra EOF

Laboratorio de programación II (Facultad de Informática)

- 5



## Ejemplo

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

int main(int argc, char* argv[])
{
    // creación del flujo ofstream y apertura del fichero
    ofstream flujoSalida( "fichero.txt", ios::out );
    // comprobar si se han producido problemas y salir del programa
    if ( !flujoSalida ) { // sobrecarga del operador !
        cerr << "el fichero no se puede abrir" << endl;
        exit( 1 );
    }
}
```

Laboratorio de programación II (Facultad de Informática)

- 6



## Ejemplo (continuación)

```
cout << "Intro edad nombre nota (Intro fin de fichero para acabar (CTRL-D)? : " << endl;

int edad;
string nombre;
double nota;

// leer datos y colocarlos en un fichero
while ( cin >> edad >> nombre >> nota ) {
    flujoSalida << edad << ' ' << nombre << ' ' << nota << endl;
    cout << " siguientes datos? \n";
}
return 0;
} // main
```

Se ha creado el fichero de texto *fichero.txt* que se puede abrir con cualquier editor de texto

Laboratorio de programación II (Facultad de Informática)

- 7



## Ejemplo escritura en fichero con objetos

```
#include .....
using namespace std;
class Persona {
private:
    int edad;
    string nombre;
    double nota;
public:
    Persona(){edad = 1; nombre= ""; nota=0.0;}
    Persona(int ed, string nom, double num){edad = ed; nombre= nom; nota=num;}
    friend ostream& operator<<(ostream&, Persona); //insertor
}; // Persona
ostream& operator<<(ostream& salida, const Persona per){
    salida << per.edad << ' ' <<per.nombre<< ' ' <<per.nota << endl;
    return salida;}

```



## Ejemplo escritura en fichero con objetos

```
int main(int argc, char* argv[])
{
    ofstream flujoSalida( "fichero.txt", ios::out );
    if ( !flujoSalida ) {
        cerr << "el fichero no se puede abrir" << endl; exit( 1 ); }
    cout << "Intro edad nombre nota: (Intro fin de fichero (CTRL-D) para acabar)? " << endl;
    int edad; string nombre; double nota;
    while ( cin >> edad >> nombre >> nota ) {
        Persona persona(edad, nombre, nota); // creación del objeto persona
        flujoSalida << persona; // escritura de la persona en el fichero
        cout << "siguientes datos?\n";
    }
    return 0;
}

```



## Ejecución y resultado

Visualización del fichero en el editor del C++ Builder



## Lectura de datos desde un fichero

```
class Persona {
private:
    int edad;
    string nombre;
    double nota;
public:
    Persona(){edad = 1; nombre= ""; nota=0.0;}
    Persona(int ed, string nom, double num){edad = ed; nombre= nom; nota=num;}
    friend ostream& operator<<(ostream&, const Persona);
    friend istream& operator>>(istream&, Persona &); //extractor
}; // Persona .....
istream& operator>>(istream& entrada, Persona &per){
    entrada >> per.edad >> per.nombre >> per.nota;
    return entrada;
}

```



## Lectura de datos desde un fichero

```
int main(int argc, char* argv[])
{
    ofstream flujoSalida( "fichero.txt", ios::out );
    if ( !flujoSalida ) { cerr << "el fichero no se puede abrir" << endl; exit( 1 ); }
    cout << "Intro edad nombre nota: (Intro fin de fichero (CTRL-D) para acabar)? " << endl;
    Persona persona;
    while ( cin >> persona ) { // uso del extractor con la entrada estándar
        flujoSalida << persona;
        cout << "siguientes datos?\n";
    }
    flujoSalida.close(); // cierre del flujo de salida
}
```



## Lectura de datos desde un fichero

```
// apertura del flujo de datos asociado a fichero.txt para entrada
ifstream flujoEntrada( "fichero.txt", ios::in );
if ( !flujoEntrada ) { // sobrecarga del operador !
    cerr << "el fichero no se puede abrir" << endl;
    exit( 1 );
}
while ( flujoEntrada >> persona ) { //lectura desde el fichero con el extractor
    cout << persona << endl;
    cout << "siguientes datos?\n";
}
return 0;
} // main
```