



UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y CC. Computación
E.T.S.I. Telecomunicación

ALMACENAMIENTO PERSISTENTE DE DATOS

Tema 1

Programación II

Tema 1: ALMACENAMIENTO PERSISTENTE DE DATOS

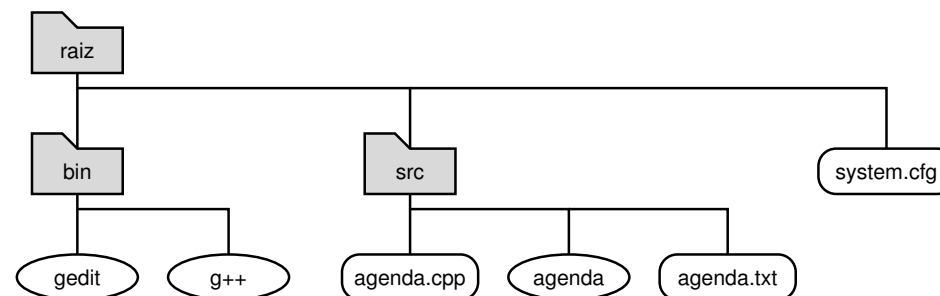
1. Introducción. Conceptos básicos
2. Entrada y Salida de Datos Asociadas a Ficheros
3. Entrada de Datos desde Ficheros de Texto
4. Salida de Datos a Ficheros de Texto
5. Ejemplo
6. Bibliografía

INTRODUCCIÓN. CONCEPTOS BÁSICOS (I)

- Almacenamiento de Datos en el Ordenador
 - Memoria Principal: (*acceso directo de la CPU*)
 - Tiempo de acceso muy rápido
 - Almacenamiento **no** persistente: *volátil*
 - Capacidad de almacenamiento limitada
 - Memoria Secundaria: (*discos duros, discos ópticos, memorias USB, etc.*)
 - Tiempo de acceso lento
 - Almacenamiento *persistente*
 - Gran capacidad de almacenamiento

INTRODUCCIÓN. CONCEPTOS BÁSICOS (II)

- Organización de la Memoria Secundaria (gran capacidad de almacenamiento)
 - Sistema de Ficheros
 - Jerarquía de Directorios (Carpetas) y Ficheros
 - Directorios: organizan jerárquicamente el sistema de ficheros
 - ◇ Directorios, subdirectorios y ficheros
 - Ficheros: almacenamiento persistente de información
 - ◇ Datos: información, configuraciones, código fuente
 - ◇ Software: bibliotecas y programas ejecutables



INTRODUCCIÓN. CONCEPTOS BÁSICOS (III)

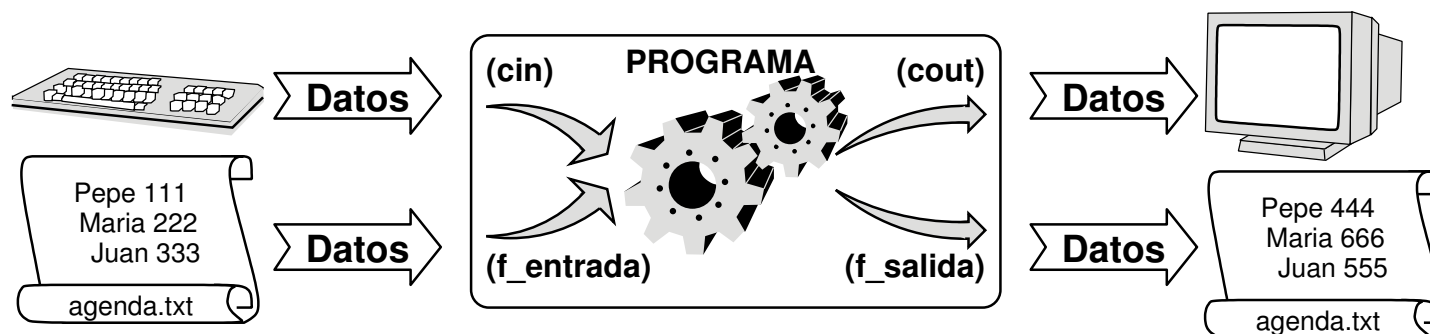
- Almacenamiento de la Información: Tipos de Ficheros
 - Ficheros de Texto
 - Codificación textual: secuencia de caracteres (ASCII) (*Conversión*)
 - Procesamiento orientado a ordenador (también humano)
 - Representan información muy diversa (editor de textos)
 - Ficheros Binarios
 - Codificación binaria: secuencia de bytes (rep. interna del ordenador)
 - Procesamiento orientado a ordenador (problemas de compatibilidad)
 - Representan información binaria (programas, imágenes, música, etc.)

Tema 1: ALMACENAMIENTO PERSISTENTE DE DATOS

1. Introducción. Conceptos básicos
2. Entrada y Salida de Datos Asociadas a Ficheros
3. Entrada de Datos desde Ficheros de Texto
4. Salida de Datos a Ficheros de Texto
5. Ejemplo
6. Bibliografía

ENTRADA Y SALIDA DE DATOS ASOCIADAS A FICHEROS (I)

- Entrada y Salida de Datos a Través de Flujos (Streams)
 - Flujos estándares de entrada y salida: *cin* y *cout*
 - Flujo de entrada: *fuentes* que proporciona una secuencia de caracteres
 - Flujo de salida: *sumidero* que recibe una secuencia de caracteres
 - Un *manejador de fichero*: *variable* que vincula un flujo de datos con un determinado fichero
 - Toda la transferencia de información se realiza a través de los manejadores de ficheros (a través de *buffers*).



ENTRADA Y SALIDA DE DATOS ASOCIADAS A FICHEROS (II)

- Procesamiento de Ficheros en un Programa C++
 1. Incluir biblioteca `<fstream>` (espacio de nombres `std`)
 2. Declarar variable (manejador de fichero) de tipo de flujo (`ifstream`, `ofstream`)
 3. Abrir el manejador de fichero: vincula el flujo con fichero (`flujo.open(...)`)
 4. Comprobar apertura correcta (`flujo.fail()`)
 5. Transferencia de información (operadores de E/S) (`>>`, `<<`, `get`, `getline`, `ws`, `ignore`)
 6. Comprobar transferencia de información correcta (`flujo.fail()`, `flujo.eof()`)
 7. Cerrar el manejador de fichero: desvincula el flujo con el fichero (`flujo.close()`)
 - Si el estado del flujo es erróneo, entonces cualquier operación de entrada o salida que se realice sobre él también fallará.
 - Paso de parámetros por **referencia** de variables manejadoras de ficheros

Tema 1: ALMACENAMIENTO PERSISTENTE DE DATOS

1. Introducción. Conceptos básicos
2. Entrada y Salida de Datos Asociadas a Ficheros
3. Entrada de Datos desde Ficheros de Texto
4. Salida de Datos a Ficheros de Texto
5. Ejemplo
6. Bibliografía

ENTRADA DE DATOS DESDE FICHEROS DE TEXTO (I)

```
#include <iostream>
#include <fstream>
using namespace std;
void leer_fich(const string& nombre_fichero, bool& ok)
{
    ifstream f_ent ;                // Manejador de Fichero de Entrada
    f_ent.open(nombre_fichero.c_str()); // Apertura del Flujo de Entrada
    if ( f_ent.fail() ) {           // ¿ Apertura Correcta ?
        ok = false ;
    } else {
        int numero ;
        f_ent >> numero ;           // Entrada de datos (Control del Buffer)
        while ( ! f_ent.fail() ) {  // ¿ Lectura Correcta ?
            cout << numero << endl ;
            f_ent >> numero ;       // Entrada de datos (Control del Buffer)
        }
        ok = ! f_ent.fail() || f_ent.eof(); // ¿ Fin de Fichero ? (End-of-File)
        f_ent.close() ;             // Cierre del Flujo de Entrada
    }
}
int main()
{
    bool ok ;
    leer_fich("datos.txt", ok) ;
    if ( ! ok ) {
        cout << "Error procesamiento de fichero" << endl ;
    }
}
```

ENTRADA DE DATOS DESDE FICHEROS DE TEXTO (II)

```

#include <iostream>
#include <fstream>
using namespace std;
void leer_fich(const string& nombre_fichero, Vector& dat, bool& ok)
{
    ifstream f_ent ; // Manejador de Fichero de Entrada
    f_ent.open(nombre_fichero.c_str()); // Apertura del Flujo de Entrada
    if ( f_ent.fail() ) { // ¿ Apertura Correcta ?
        ok = false ;
    } else {
        int numero ; ok = true ;
        f_ent >> numero ; // Entrada de datos (Control del Buffer)
        while ( ! f_ent.fail() && ok ) { // ¿ Lectura Correcta ?
            insertar(dat, numero, ok) ;
            f_ent >> numero ; // Entrada de datos (Control del Buffer)
        }
        ok = ok && ( ! f_ent.fail() || f_ent.eof() ); // ¿ Fin de Fichero ? (End-of-File)
        f_ent.close() ; // Cierre del Flujo de Entrada
    }
}
void insertar(Vector& dat, int num, bool& ok)
{
    if (dat.nelms == int(dat.elm.size())) {
        ok = false ;
    } else {
        ok = true ;
        dat.elm[dat.nelms] = num ;
        ++dat.nelms ;
    }
}
int main()
{
    bool ok ;
    Vector dat ; dat.nelms = 0 ;
    leer_fich("datos.txt", dat, ok) ;
    if ( ! ok ) {
        cout << "Error" << endl ;
    } else { ... }
}

```

Tema 1: ALMACENAMIENTO PERSISTENTE DE DATOS

1. Introducción. Conceptos básicos
2. Entrada y Salida de Datos Asociadas a Ficheros
3. Entrada de Datos desde Ficheros de Texto
4. Salida de Datos a Ficheros de Texto
5. Ejemplo
6. Bibliografía

SALIDA DE DATOS A FICHEROS DE TEXTO (I)

```
#include <iostream>
#include <fstream>
using namespace std;
void escribir_fich(const string& nombre_fichero, bool& ok)
{
    ofstream f_sal ;                // Manejador de Fichero de Salida
    f_sal.open(nombre_fichero.c_str()); // Apertura del Flujo de Salida
    if ( f_sal.fail() ) {           // ¿ Apertura Correcta ?
        ok = false ;
    } else {
        int numero ;
        cin >> numero ;
        while ((numero > 0) && ! cin.fail() && ! f_sal.fail()) { // ¿ Flujo Correcto ?
            f_sal << numero << endl ;                          // Salida de datos (SEPARADORES ADECUADOS)
            cin >> numero ;
        }
        ok = ! f_sal.fail() ;           // ¿ Salida Correcta ?
        f_sal.close() ;                // Cierre del Flujo de Salida
    }
}
int main()
{
    bool ok ;
    escribir_fich("datos.txt", ok) ;
    if ( ! ok ) {
        cout << "Error escritura de fichero" << endl ;
    }
}
```

SALIDA DE DATOS A FICHEROS DE TEXTO (II)

```
#include <iostream>
#include <fstream>
using namespace std;
void escribir_fich(const string& nombre_fichero, const Vector& dat, bool& ok)
{
    ofstream f_sal ;                // Manejador de Fichero de Salida
    f_sal.open(nombre_fichero.c_str()); // Apertura del Flujo de Salida
    if ( f_sal.fail() ) {           // ¿ Apertura Correcta ?
        ok = false ;
    } else {
        for (int i = 0; (i < dat.nelms) && ! f_sal.fail(); ++i) { // ¿ Hay Datos ? ¿ Flujo Correcto ?
            f_sal << dat.elm[i] << endl ;                          // Salida de datos (SEPARADORES ADECUADOS)
        }
        ok = ! f_sal.fail() ;    // ¿ Salida Correcta ?
        f_sal.close() ;          // Cierre del Flujo de Salida
    }
}
int main()
{
    bool ok ;
    Vector dat;
    // actualizar dat con datos adecuados
    escribir_fich("datos.txt", dat, ok) ;
    if ( ! ok ) {
        cout << "Error" << endl ;
    } else { ... }
}
```

Tema 1: ALMACENAMIENTO PERSISTENTE DE DATOS

1. Introducción. Conceptos básicos
2. Entrada y Salida de Datos Asociadas a Ficheros
3. Entrada de Datos desde Ficheros de Texto
4. Salida de Datos a Ficheros de Texto
5. Ejemplo
6. Bibliografía

EJEMPLO

- Programa que gestione una agenda personal (nombres y teléfonos de personas) y permita tanto guardar como cargar los datos a/desde un fichero. El programa mostrará un menú para seleccionar las acciones deseadas.

Tema 1: ALMACENAMIENTO PERSISTENTE DE DATOS

1. Introducción. Conceptos básicos
2. Entrada y Salida de Datos Asociadas a Ficheros
3. Entrada de Datos desde Ficheros de Texto
4. Salida de Datos a Ficheros de Texto
5. Ejemplo
6. Bibliografía