



Departamento de Lenguajes
y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

LABORATORIO DE PROGRAMACIÓN

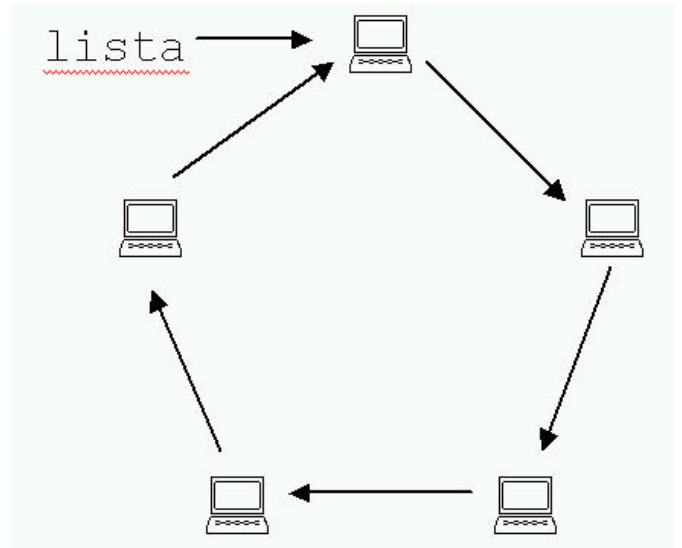
E.T.S.I.Informática- GESTIÓN. 2-SEPTIEMBRE-2004

Duración del Examen: 4 horas

APELLIDOS _____ NOMBRE _____

ORDENADOR _____ GRUPO (A/B/C) _____

Se desea realizar una aplicación para la gestión de mensajes en una red de computadores que están conectados en forma de anillo unidireccional (tal como se muestra en la figura) y que será modelada mediante una lista circular cuyos nodos están ordenados por su dirección física. Cada nodo de la lista representa un ordenador y se tendrá para cada ordenador la siguiente información: nombre simbólico (cadena de caracteres), dirección física (número natural), sistema operativo (un enumerado de valores : Windows, Unix, MacOS, DOS) y una lista de un máximo de 10 mensajes almacenados (cada mensaje será una cadena de caracteres).



Se pide que se implemente un módulo llamado MOrdenador (MOrdenador.h , MOrdenador.cpp), un módulo llamado MListaC (MListaC.h , MListaC.cpp) con las operaciones que se indican más abajo y un programa llamado correo.cpp con las siguientes opciones de menú:

Nombre: (Apellidos, Nombre)	Curso: 1º
Especialidad: (Gestión/Sistemas)	Grupo: A/B/C
Puesto: número de ordenador	Fecha: 02/09/04

MENU PRINCIPAL

=====

- A. Insertar Un Ordenador en la Red.
- B. Leer los Mensajes de Un Ordenador en la Red.
- C. Mandar un Mensaje a Un Ordenador en la Red.
- D. Borrar Un Ordenador de la Red.
- E. Cargar Estado de la red desde Fichero.
- F. Salvar Estado de la red a Fichero.

X. Salir del Programa

Introduzca Opción:

Opciones:

A. Insertar Un Ordenador en la Red. Se solicitará al usuario los valores del nombre, dirección y sistema operativo y se insertará en la estructura. Al introducirlo como ordenador nuevo no tendrá ningún mensaje. Caso de que ya hubiera otro con la misma dirección IP se ignorará la inserción y se informará del error. Ejemplo:

```
Nombre:          pc125
Dirección IP:    8190
S.O.:           windows
```

B. Leer los Mensajes de Un Ordenador en la Red. Se solicitará al usuario la dirección de la máquina y se mostrará por pantalla todos los mensajes almacenados en la misma. Si la máquina no existiera se informará del error. Ejemplo:

```
Introduzca Dirección IP: 8190
Lista de Mensajes
```

```
-----
```

1. Pepe ha salido a comer.
2. La Máquina pc125 no funciona.

C. Mandar un Mensaje a Un Ordenador en la Red.

Se solicitará al usuario la dirección de la máquina y el mensajes a enviar. Si la máquina no existiera se informará del error. Ejemplo:

```
Introduzca Dirección IP: 8190
Mensaje: Esta es una prueba de la opción C.
```

D. Borrar Un Ordenador de la Red. Se solicitará al usuario la dirección de la máquina y se borrará de la lista. Si la máquina no existiera no se hará nada. Ejemplo:

```
Introduzca Dirección IP: 8190
Borrada ...
```

E. Cargar Estado de la red desde Fichero. Se solicitará al usuario el nombre de un fichero de texto con el formato adjunto y se añaden todas las máquinas que no estuvieran en la red con sus respectivas listas de mensajes.

Formato del Fichero:

```
<nombre><SP><dir><SP><SO><SP><NUM_MSJ><SP><MSJ1>#<MSJ2>#...<MSJN><ENTER>
```

donde:

```
<nombre>  Nombre de la máquina.
<dir>     Dirección de la máquina.
<SP>     Espacio en blanco
<SO>     Sistema operativo de la máquina.
<NUM_MSJ> Número de mensajes almacenados en la máquina.
<MSJi>   Mensaje i-ésimo de la máquina.
```

Fichero de Ejemplo "red.txt"

```
pc121 111 Windows 2 Este es el Mensaje 1# y este el Mensaje 2
pc122 222 Unix 1 Hola a todos
pc123 333 MacOS 0
pc124 444 DOS 3 cambiadme el SO# no te lo cambio# que pena
```

Ejemplo del Funcionamiento de la Opción:

Nombre Fichero: red.txt

F. Salvar Estado de la red a Fichero. Se solicitará al usuario el nombre de un fichero de texto y se grabará todo el contenido de la red con el mismo formato de lectura. Si el fichero ya existiera se borrará su contenido. Ejemplo:

Nombre Fichero: red.txt

X. Salir del Programa. En esta opción se pedirá confirmación de salida. Si es afirmativa, se terminará el programa, liberando todos los recursos que hayan sido reservados y si es negativa, se volverá al menú principal.

Después de cada opción se hará una pausa y se limpiará la pantalla antes de volver al menú principal.

Módulos a Implementar, además del programa principal (correo.cpp):

❖ Módulo MOrdenador (MOrdenador.h, MOrdenador.cpp):

➤ Define las **Constanes**:

- MAXCAD. Tamaño máximo de las cadenas a 40.
- MAXMSJ. Máximo número de mensajes a 10.
- FINCAD. Finalizador del cadenas.
- ENTER.
- SEP. Separador de mensajes en fichero como '#'
- SP. Espacio en Blanco como ' '

➤ Define los **Tipos**:

- TNatural. Entero sin signo.
- TCadena. Cadena de Caracteres.
- TSO. Enumerado con los distintos sistemas operativos.
- TMensaje. Datos de losMensajes.
- TOrdenador. Datos de un ordenador.

➤ Define los **Procedimientos y Funciones**:

- IgualCadena. Nos dice si 2 cadenas son iguales.
- CopiaCadena. copia una cadena origen en otra destino.
- Cadena_a_TSO. Pasa una cadena de caracteres al enumerado de tipo TSO al que representa.
- TSO_a_Cadena. Pasa un enumerado de tipo TSO a la cadena de caracteres que lo representa.
- LeerOrdenador. Lee toda la información de un ordenador (sin mensajes) desde teclado.
- LeerOrdenadorFichero. Lee toda la información de un ordenador (con mensajes) desde un fichero del cual nos dan su manejador.
- EscribirOrdenador. Escribe por pantalla toda la información de un ordenador (con mensajes).
- EscribirOrdenadorFichero. Escribe toda la información de un ordenador (con mensajes) a un fichero del cual nos dan su manejador.
- InsertarMensaje. Añade un mensaje a los que ya hay en un ordenador.
- ObtenerMensaje. Extrae de un ordenador el mensaje que lleve más tiempo almacenado.

- ❖ Módulo MListaC(MListaC.h, MListaC.cpp): Módulo de manejo de una LISTA ENLAZADA CIRCULAR Y ORDENADA por valores de dirección IP.
 - Define los **tipos**
 - TListaC
 - TNodo.
 - Define los **Procedimientos y Funciones:**
 - CrearListaC: Crea una TListaC vacía
 - ListaCVacia: Nos dice si una TListaC está vacía
 - ListaCLlena: Nos dice si una TListaC está llena
 - InsertarListaC: Inserta un Ordenador en la TListaC
 - EliminarListaC: Elimina (si existe) el ordenador cuya dirección IP se indica de la TListaD
 - BuscarListaC. Retorna (si existe) el Ordenador cuya IP coincida en la TListaC
 - RangoDireccionesListaC. Retorna el menor y el mayor valor de las IP's
 - DestruirListaC. Destruye una TListaC.

EL AÑADIR OPERACIONES EN LA PARTE DE DEFINICIÓN DE ALGÚN MÓDULO, ASÍ COMO EL USO DE DETALLES DE IMPLEMENTACIÓN DE LOS MODULOS DENTRO DEL PROGRAMA PRINCIPAL SERÁ CAUSA DE SUSPENSO (ES DECIR, ESTÁ PROHIBIDO USAR 'l->', NULL, ETC FUERA DE LA IMPLEMENTACIÓN DEL MÓDULO DE LISTAS).

Notas:

1. Todas las cadenas de caracteres tendrán un máximo de 40 caracteres.
2. Es obligatorio trabajar en el directorio **C:\LPSEPG**. Si no existe se creará
3. Para **Aprobar** deberá ser correcta la definición de tipos, la modularización y funcionar **CORRECTAMENTE** las Opciones A,B,C, D, E y X del menú.
4. Se recomienda y valora la buena descomposición del programa principal en procedimientos y funciones, así como el uso de procedimientos y funciones auxiliares dentro de la implementación de los módulos cuando estas sean necesarias.
5. Recordad que en el nuevo compilador cambian las cabeceras para las bibliotecas estándar y las de entrada/salida.

Ejemplo:

```
// Incluir E/S y Librerías Standard
#include <iostream>
#include <cstdlib>
using namespace std;
```

6. Recordad que todo fichero de cabecera debe llevar su pragma #ifndef #endif

Ejemplo: Fichero: cabecera.h

```
#ifndef _Cabecera_h_
#define _Cabecera_h_
// Contenido del archivo de Cabecera
#endif
```