Universidad de Málaga
E.T.S.I. Telecomunicación

Laboratorio de Programación 2
septiembre de 2003
(1^{o} de Ingeniería de Telecomunicación)

Alumno: Grupo: Máquina:

Se desea implementar un servidor de telefonía móvil. El servidor almacenará, para cada teléfono móvil que se encuentre en cobertura, una estructura `TMovil` con los siguientes campos:

número El número del móvil se representará por un valor del tipo `TNumMovil`. Por simplicidad, este tipo se define como `unsigned long int`.

estado Un móvil puede encontrarse en tres estados: disponible, llamando y llamado. Estos estados se representarán por el tipo enumerado `TEstMovil`.

conectado_con En caso de que un móvil no se encuentre disponible, este campo almacena el número del móvil (`TNumMovil`) con que se está conversando.

Para mejorar el rendimiento, el servidor almacenará la información de los móviles en un array de N listas. La lista (es decir, la posición del array) que corresponde a un móvil se calcula tomando el resto de dividir el número del móvil, M, entre la dimensión del array, N; es decir, al móvil M le corresponde la lista M mod N. Además, cada lista estará ordenada crecientemente por el número de móvil.

La figura de abajo muestra la estructura del servidor para un array de dimensión N= 5. En este caso, hay tres móviles en cobertura: 16, 71 y 28. A los dos primeros les corresponde aparecer en la lista 1, pues 16 mod 5 = 71 mod 5 = 1. Al móvil 28 le corresponde la lista 3 (28 mod 5 = 3). Observa que los móviles de la lista 1 aparecen ordenados crecientemente. El estado de cada móvil se representa en la figura por una letra. El móvil 71 está disponible (`D`), el móvil 16 está llamando (`L`) al 28, que está siendo llamado (`R`).

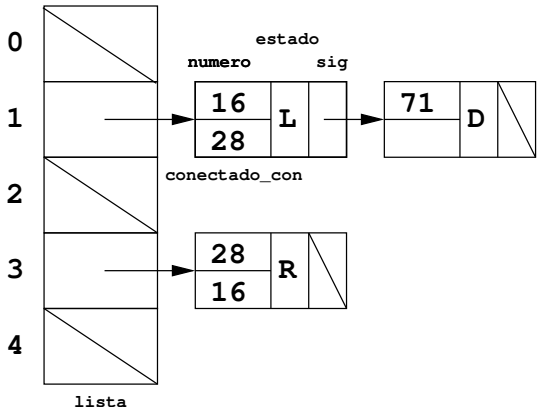

Figura 1: Servidor de telefonía atendiendo a tres móviles

La implementación del servidor se realizará a través de la clase `CServidor`, definida por el siguiente interfaz:

```
#ifndef __CSERVIDOR__
#define __CSERVIDOR__

class CServidor {

   public:
```

```
      // tipos publicos

      typedef unsigned long int TNumMovil;
      enum TEstMovil {disponible, llamando, llamado};
      enum TRes {conectado, sin_cobertura, comunicando}; // resultado de una llamada

    // metodos publicos

    CServidor();
    ~CServidor();

    TEstMovil estado_movil(const TNumMovil n) const;
      // devuelve el estado en que se encuentra el movil n

    void entra_en_cobertura(const TNumMovil n);
      // se inserta el movil n en el servidor, con estado disponible

    void sale_de_cobertura(const TNumMovil n);
      // se elimina el movil n del servidor.si estaba comunicando con otro movil,
      // debe inicializarse el estado de este a disponible

    void  llamar(const TNumMovil orig, const TNumMovil dest, TRes& res);
      // se establece una llamada entre los moviles orig y dest. el parametro de salida
      // res indica el resultado de la llamada, de tipo TRes

    void colgar(const TNumMovil n);
      // el movil n y aquel con que este conectado vuelven a estar disponibles

    void llamadas_en_curso(ostream& listado) const;
      // imprime en un fichero un listado con las llamadas en curso, indicando el origen
      // y destino de cada llamada

  private:

      struct TMovil {
          TNumMovil  numero;
          TEstMovil  estado;
          TNumMovil  conectado_con;
      };

      struct TNodo {
         TMovil movil;
         TNodo* sig;
      };

      typedef TNodo* TNodoP;

      static const int N= 5;

      TNodoP lista[N];
};

#endif
```

Notas

- Para aprobar es necesario hacer los algoritmos: `CServidor`, `~CServidor`, `estado_movil`, `entra_en_cobertura`, `sale_de_cobertura`.
- En el directorio `Z:\etsit\lp2sep03` está disponible el fichero `servidor.hpp`.
- El directorio de trabajo debe ser `C:\lp2tel`.