Práctica 5:
Gestor de correo *fdimail* 2.0

Fecha de entrega: 31 de mayo de 2015

1. Descripción

Se ha observado una limitación muy importante en la versión actual del gestor de correo que hace imposible su uso en un contexto realista. Concretamente, tanto el número de usuarios registrados en el sistema, como los números de correos mantenidos por el gestor y registros de las bandejas de correo, están muy limitados. Este límite además es bastante bajo pues al utilizarse solamente memoria de pila, ésta se desborda con facilidad. Se deberán por tanto re-implementar las estructuras de datos causantes del problema de manera que desaparezca dicha limitación.

2. Detalles de implementación

Para poder manejar listas más grandes, hay que emplear estructuras de datos mejoradas que hagan uso de la memoria dinámica para almacenar los datos. En concreto, se emplearán datos y arrays dinámicos de distintos tamaños. A continuación se indican las pautas a seguir para los distintos tipos de listas:

tListaCorreos y tListaRegistros

Se deben utilizar arrays dinámicos (Tema 9), pero optimizando la gestión de memoria: en lugar de usar un mismo array dinámico durante toda la ejecución, se irá modificando el tamaño del array para adaptarlo a las necesidades de cada momento, siguiendo esta política:

- Cuando se cree una lista nueva (al arrancar el gestor sin ficheros), se creará el array dinámico con un tamaño de 10.
- Al cargarse las listas desde fichero, se crearán los arrays con una capacidad suficiente como para guardar todos los elementos, redondeado a la siguiente decena. Para ello, el fichero de correos deberá comenzar por un número entero que indique el número de correos existentes. Por ejemplo, si el fichero de correos comienza por un 12, se creará un array con capacidad para 20 correos. El centinela del fichero de correos ya no será necesario y debe quitarse.
- Si al añadir un elemento el array ya estuviese lleno, se ampliará su tamaño según la siguiente fórmula: $nuevaCapacidad = (viejaCapacidad * 3) / 2 + 1$.

tListaUsuarios

En este caso, se hará uso de un array estático de datos dinámicos, es decir, un array de punteros a tUsuario. El tamaño del array vendrá determinado por la constante MAX_USUARIOS cuyo valor por defecto será 50.

3. Partes opcionales

3.1 Reducción del tamaño de tListaRegistros

Modificar el comportamiento del array dinámico de tListaRegistros para que además de aumentar su tamaño también lo reduzca cuando sea necesario para optimizar aún más el uso de la memoria.

Cada vez que se elimine un registro de la lista de registros, si en el array dinámico quedan más de la mitad de las posiciones sin ocupar se reducirá el tamaño del array dinámico en 1/3, con lo que la fórmula será: *nuevaCapacidad = viejaCapacidad * 2/3.*

3.2 tRegistro con punteros

Con el objetivo de evitar las frecuentes búsquedas en la lista de correos desde los registros de correo, se propone cambiar el tipo tRegistro de manera que en lugar de guardar el identificador de correo guarde directamente un puntero a éste. Durante el proceso de carga de la lista de registros, deberán obtenerse los punteros a los correos buscándolos por identificador en la lista de correos (que por tanto debe cargarse antes).

3.3 tListaCorreos como array dinámico de datos dinámicos

Además, para evitar complicaciones en una hipotética eliminación de correos de la lista de correos (parte opcional de la práctica 4), la lista de correos deberá implementarse como un array dinámico de datos dinámicos, es decir un array dinámico de punteros a tCorreo.

4. Entrega de la práctica

La práctica se entregará a través del Campus Virtual. Se habilitará una nueva tarea: **Entrega de la Práctica 5** que permitirá subir un fichero comprimido Practica5GrupoXX.zip que incluya todos los ficheros fuente del proyecto (.h y .cpp). Uno sólo de los miembros del grupo (no los dos) será el encargado de subir la práctica.

Fin del plazo de entrega: **31 de mayo a las 23:55**