PRÁCTICA 2: Lectura de ficheros y uso de listas

OBJETIVOS

El objetivo de esta práctica es implementar un TAD Base de Datos que ofrezca una serie de servicios básicos para gestionar la base de datos de premios y premiados. Para ello se implementarán funciones de lectura de archivos que utilizarán los TAD desarrollados en la práctica anterior para mantener los datos leídos en memoria. Asimismo, se implementarán funciones de consulta.

Además de la documentación que los entornos de desarrollo proporcionan sobre el lenguaje de programación C, mediante la página de Moodle de la asignatura los profesores de los laboratorios proporcionarán las indicaciones adicionales para desarrollar el trabajo de cada una de las sesiones.

NORMATIVA DE ENTREGA

Como resultado de esta práctica debe entregarse mediante la herramienta Moodle, en la actividad que los profesores de los laboratorios indiquen, un fichero empaquetado (.zip) que contenga todos los ficheros que permitan abrir, construir el ejecutable y ejecutar su proyecto **antes de la primera sesión de la práctica siguiente. Cada grupo puede entregar hasta las 12 de la noche del día anterior al que su profesor iniciará la siguiente práctica.** Este fichero empaquetado debe, por tanto, contener al menos los siguientes ficheros:

- README.txt: fichero de texto plano donde han de constar vuestros datos (nombres, e-mail, grupo de prácticas, número de grupo) y cualquier otra indicación adicional sobre la práctica. Sobre la descripción de vuestro trabajo es suficiente con que expliquéis las cuestiones que os hayan podido parecer relevantes de vuestra solución. No se trata de que repitáis las instrucciones del enunciado ni las indicaciones que vuestros profesores hayan hecho en clase. Sólo debéis centraros en las peculiaridades que vuestra solución contenga. Añade en este fichero de forma clara la lista de comandos necesarios para generar los ejecutables.
- LISTA DE FICHEROS
    - premiado.h : Cabecera del TAD Premiado.
    - premiado.c : Implementación del TAD Premiado.
    - premiados.h: Cabecera del TAD Premiados.
    - premiados.c: Implementación del TAD Premiados.
    - premio.h: Cabecera del TAD Premios.
    - premio.c: Implementación del TAD Premios.
    - premios.h: Cabecera del TAD Premios.
    - premios.c: Implementación del TAD Premios.
    - lista.h
    - lista.c

- data_base.h: Cabecera del TAD base de datos
- data_base.c: Implementación del TAD base de datos.
- const.h: Fichero de cabecera con constantes globales de la aplicación.
- main_p2.c: Programa principal que incluye toda la funcionalidad de la práctica.

Se debe entregar todo en un zip.

El nombre del fichero comprimido tiene que seguir la siguiente estructura:
lab<turno>_gr<numero_pareja>_p2.zip

Donde:

- <turno> es el turno asignado (4111, 4112, 4113 o 4114)
- <numero_pareja> es el que identifica a tu pareja dentro de tu laboratorio

Es obligatorio respetar todos aquellos nombres y formatos que se describen en el enunciado.

¡No olvidéis incluir vuestros nombres en los ficheros entregados!

CRITERIOS BÁSICOS DE EVALUACIÓN

Las siguientes normas deben ser consideradas en la realización de los problemas propuestos y la entrega del material:

1. *Estilo de programación*: debe ajustarse a las normas básicas de estilo de la programación estructurada en C y que ya se han presentado en asignaturas anteriores de programación.
2. *Documentación del código*: todas las funciones que se programen deberán estar conveniente y brevemente comentadas, tanto en su cabecera (funcionalidad y parámetros de entrada/salida), como en su cuerpo (descripción de etapas relevantes). Se evitarán comentarios obvios o redundantes.
3. *Pruebas*: es especialmente importante el control de errores. Es justificable que un programa no admita valores muy grandes de los datos, pero no es justificable que el programa tenga un comportamiento anómalo con dichos valores.
4. *Comprobación y propagación de errores*: se deberán comprobar todos los errores susceptibles de producirse en el programa, incluyendo ausencia de archivo de datos de entrada, errores de formato en datos de entrada, fallo de funciones del sistema operativo (malloc, etc.). Se considerará fallo grave que el programa aborte su ejecución por un tratamiento de errores insuficiente o inadecuado.
5. *Liberación de memoria dinámica*: al finalizar los programas, éstos deberán haber liberado toda la memoria dinámica que hayan reservado durante su ejecución.
6. *Uso de Valgrind*. Es imprescindible que todos los ejercicios hayan sido probados usando Valgrind para asegurarse de que no existen problemas de memoria.

INTRODUCCIÓN

Para realizar la práctica se proporcionará una base de datos compuesta de dos ficheros que contienen premiados y premios. Estos ficheros siguen el formato que se describe a continuación.

- fpremiados.csv
  Este fichero incluye la información relativa a los premiados, con una línea por cada premiado de la base de datos. El fichero contiene los siguientes campos separados por comas (","): Identificador numérico, nombre, país y género del premiado.

  p.e..:

  ```
  6,Marie Curie,FR,female
  5,Pierre Curie,FR,male
  881,European Union,EU,org
  ```

- fpremios.csv
  Este fichero incluye la información relativa a los premios, con una línea por cada premio en la base de datos. Los campos incluidos en el fichero están separados por comas y son los siguientes: identificador, año, categoria, descripción del premio e identificadores de los premiados separados por barras "|".

  p.e.:

  ```
  1,1906,fisica,descripcion premio,6|5
  2,2012,paz,descripcion premio,881
  ```

1ª SEMANA: TAD Base de Datos - Funciones de lectura

Se implementará el TAD DataBase, que se encargará de la gestión. La estructura de datos y las operaciones se almacenarán en los ficheros data_base.h y data_base.c.
En esta primera semana se propone crear la estructura para almacenar las colecciones de premiados y premios, así como las funciones relativas a la lectura y escritura de los ficheros de datos.

- Crear la base de datos; la función toma como parámetro el directorio donde se encuentran los ficheros de datos de la base de datos.

```
data_base *new_data_base(char *dir_name);
```

- Eliminar la base de datos y liberar sus recursos:

```
void destroy_data_base(data_base *bd);
```

Se recomienda crear las siguientes funciones auxiliares para apoyar la lectura de los datos. Observa que estas funciones son auxiliares, es decir, su utilidad es facilitar la implementación del TAD pero su propósito no es que sean invocadas desde fuera del TAD. Por tanto, no deben aparecer en el fichero de cabecera.

- Función auxiliar para la lectura de los datos de un premiado. Esta función recibe un puntero a una cadena de caracteres y, comprueba que sigue el formato adecuado, devuelve un puntero a TAD premiado.
    - Es importante tener en cuenta que hay que eliminar el posible retorno de carro de la cadena. Debe decidirse si esta operación se realiza en esta función o si es responsabilidad del código que realiza la llamada a la función. En cualquier caso, lo importante es documentar la decisión en la documentación de la función.

```
premiado *bd_parse_premiado(char *str);
```

- Función auxiliar para leer un fichero con datos de premiados. Crea la estructura *premiados* y la rellena utilizando la información del fichero. Devuelve NULL si hay algún error al manejar el fichero.

```
premiados *bd_parse_file_premiados(char *file_name);
```

- Función auxiliar similar a la de lectura de premiado, pero para la lectura de los datos de un premio. Esta función recibe un puntero a una cadena de caracteres, comprueba que sigue el formato adecuado, y devuelve un puntero al TAD *premio*. Además recibe la lista de premiados para enlazar cada premio con sus premiados.

```
premio *bd_parse_premio(char *str, premiados *aa);
```

- Función auxiliar para leer un fichero con datos de premios. Crea la estructura *premios* y la rellena utilizando la información del fichero. Devuelve NULL si hay algún error al manejar el fichero. Además recibe la lista de premiados para enlazar cada premio con sus premiados.

```
premios *parse_file_premios(char *file_name, premiados *aa);
```