Actividad Grupal: Recursividad y ficheros

**Objetivos**

* Aprenderás el concepto de recursión aplicado al diseño de algoritmos.
* Comprenderás las ventajas e inconvenientes de la recursión frente a las versiones iterativas.
* Conocerás los tipos de recursión.
* Podrás diseñar e implementar funciones recursivas en C
* Aprenderás los tipos de ficheros en lenguaje C.
* Realizarás ejercicios relacionados con operaciones en ficheros en C.

**Descripción de la actividad**

Realiza los siguientes programas en C:

Ejercicio 1

Implementa un algoritmo de tipo iterativo para calcular la serie de Fibonacci y a continuación construir una función recursiva para resolver la serie de Fibonacci.

Fib(0) = Fib(1) = 1

Fib(n) = Fib(n − 1) + Fib(n − 2)

**Resolución del Ejercicio 1**

El primer paso que tomamos para resolver el ejercicio de la creación e implementación del algoritmo iterativo de Fibonacci creando una función recursiva que fuera llamada en cuanto un usuario nos solicitase el número que correspondía en la serie a una posición concreta, fue crear el Algoritmo de la sucesión Fibonacci.

Para ello, y dado que estamos hablando de que podíamos tener valores altos, definimos primero la variable como de tipo long (finalmente no utilizamos de tipo unsigned long pero hubiese tenido más sentido porque el Dominio de la serie esta en N) con tamaño 32 bits. Lo siguiente era utilizar el metodo burbuja aprendido en ejercicios anteriores para ir iterando por la formula y guardando los resultados en espacio de memoria auxiliar hasta retornar el resultado.

Tras esto definimos la función recursiva, que se iba a llamar a si misma para resolver la sucesión según la posición requerida por el usuario. Con un bucle if donde evitamos los calculos de las primeras dos posiciones (0,1) donde los resultados son los mismos valores, pasamos a las llamadas a la función recursiva en la posicion -1 y -2.

Tras esto, simplemente montamos la función main para pedirle al usuario la posición requerida y llamar a la función recursiva.

Hay que añadir, que en esta función hemos limitado la posición que nos solicita el usuario para que los calculos que debe realizar el

procesador no desborden la pila (StackOverFlow) o conlleve un tiempo excesivo de procesamiento.

Ejercicio 2

Implementa un algoritmo que se encarguen de abrir, leer, almacenar y modificar la información de un archivo secuencial en el que se almacena información de los libros de una biblioteca. La información de cada uno de los libros que se tiene que almacenar es:

* Título.
* Autor.
* Editorial.
* Año de edición.
* ISBN.
* Número de páginas.

**Resolución del Ejercicio 2**

Lo primero que debemos generar para resolver este ejercicio es la definición de una variable global con el archivo destino y la estructura de datos solicitada.

Algunas de las variables las hemos creado como arrays con un tamaño concreto para manejar mejor la interacción con el usuario que va a meter los datos solicitados.

Tras esto, primero incluimos las funciones principales que van a ser aquellas que realicen las acciones solicitadas, y luego las funciones que manejaran las funciones principales para obtener los resultados.

Hemos limpiado de tal manera la función main, que solo dispone de tres lineas de código con una función principal y la de retorno.

Las funciones principales junto con las de manejo de estas son las que mas tiempo nos ha llevado, la más sencilla fue la de mainMenu() ya que seguía el patrón de ejercicios anteriores.

Tras esta, definimos showMenu(), que llamaba a su vez a la función getBook(). Las funciones están compuestas por un array definido en base a la estructura de datos original, la primera aparte de pasar una validación de si existe algún libro o no, realiza un bucle para imprimir los datos solicitados de cada libro. La segunda, algo más compleja, es la que con las funciones de fseek() y fread(), se posiciona en el archivo y lo lee para encontrar los libros incluidos.

La siguiente función es insertMenu(), que llama primero a la función validateBook(), para manejar si ya existe la referencia introducida por el usuario, y más tarde la de insertBook(). La primera basicamente con la función fread() recorre las referencias existentes y la segunda utiliza fwrite() para introducir los datos que le estamos solicitando al usuario.

La función deleteMenu() o la acción de eliminar referencia de la lista de libros no estaba solicitada como tal en el ejercicio, y la hemos incluido dado el valor adicional que le aporta al usuario poder eliminar registros. Esta función además de volver a validar si la referencia del libro existe o no, se maneja con función deleteBook() que es la realmente se encarga de eliminar el registro del fichero. En esta función abrimos el archivo como en el resto con fopen() y lo leemos con fread(), pero la verdadera magia de la función viene cuando se localiza la referencia, se posiciona en el archivo y posiciona el puntero de tal manera que restamos el registro.

La última función es quizás la más compleja en cuanto a desarrollo, no esta resuelta todo lo bien que nos hubiera gustado porque entre otras cosas cuando el usuario quiere modificar un registro tiene que volver a meter todos los datos uno a uno. Al tener varias dependencias no hemos podido dejarla como nos gustaría, pero resuelve el ejercicio y la necesidad del usuario de manera adecuada. De nuevo la función modifyMenu() llama a otra que realmente maneja la modificación de los registros que es modifyBook(), el funcionamiento de esta última sigue en cierta el patrón de anteriores funciones aunque utilizar un nuevo objeto (booky) para sustituir lo previamente ingresado.

El programa debe mostrar el nombre y DNI de los componentes del grupo en la primera línea de la pantalla.

Una vez acabada la actividad, la entrega correcta del trabajo realizado consiste en la realización de un documento (utilizando la plantilla de Microsoft Word disponible en el aula virtual) explicando la realización de los diferentes ejercicios:

* Cómo se llega a la solución.
* Tipos de estructuras utilizadas.
* Justificación del uso de dichas estructuras.
* Comentarios sobre los posibles errores o problemas que hayas encontrado a la hora de la realización.

Además se adjuntará el código fuente final de la solución de cada uno de los problemas y el ejecutable final generado, donde aparecerá en pantalla el nombre del grupo.

En todos los archivos debe de quedar claramente definido qué es cada archivo. Por ejemplo, el ejecutable del problema número 1 se llamará: *Nombre\_de\_grupo\_1*. Una vez finalizada la actividad, todos los archivos deberán ser incluidos en un archivo comprimido (.zip) con el nombre de grupo, por ejemplo: *Nombre\_de\_grupo.zip*.

El trabajo consistirá en realizar un documento (1 página de portada, una página de índice y seis páginas) donde se verá el código del programa realizado y la explicación de la solución. Además del documento se adjuntarán los archivos generados por el compilador de C.

**Rúbrica**

|  |  |  |  |
| --- | --- | --- | --- |
| Recursividad y ficheros  (valor real: 4 puntos) | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | El código compila | 4 | 40% |
| Criterio 2 | La estructura y diseño es adecuada | 2 | 20% |
| Criterio 3 | Los resultados devueltos son correctos | 3 | 30% |
| Criterio 4 | El programa es óptimo y está correctamente comentado | 1 | 10% |
|  |  | **10** | **100 %** |

**Extensión** **máxima:** la memoria deberá tener un máximo de 10 páginas, fuente Calibri 12 e interlineado 1,5.

**Organización y gestión de equipos**

* Coordinación Académica organizará los equipos y publicará en el foro de la asignatura la composición de los mismos.
* Para trabajar en equipo cada estudiante dispondrá de la cuenta de correo de Office 365 de la que dispone (@comunidadunir.net).
* Una vez realizado el trabajo, cada uno de los miembros de un mismo equipo subirá el mismo documento a la tarea correspondiente.
* Evaluación del trabajo en grupo. Piensa de forma global en todos los compañeros del grupo y valora estas preguntas. La entrega consistirá en la siguiente tabla realizada individualmente más el trabajo grupal de la actividad colaborativa.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sí | No | A veces |
| Todos los miembros se han integrado al trabajo del grupo | x |  |  |
| Todos los miembros participan activamente | x |  |  |
| Todos los miembros respetan otras ideas aportadas | x |  |  |
| Todos los miembros participan en la elaboración del informe | x |  |  |
| Me he preocupado por realizar un trabajo cooperativo con mis compañeros | x |  |  |
| Señala si consideras que algún aspecto del trabajo en grupo no ha sido adecuado |  | x todo bien |  |

En el canal de UnirTV puedes encontrar tutoriales de las distintas herramientas:

<http://tv.unir.net/secciones/3967/4883/0/0/0/0/>