

MEMORIA EDAT PRÁCTICA 6

Ignacio Campos y Alberto Castro

El objetivo de la práctica es implementar una base de datos. En ella, se mantendrá un índice en memoria dinámica. Para su funcionamiento, se implementará las funciones de búsqueda, borrado e inserción con sus respectivas estrategias como son:

- First fit: inserta el registro en el primer hueco que cabe.
- Worst fit: inserta el registro en el hueco más grande.
- Best fit: inserta el registro en el hueco más pequeño siempre y cuando quepa.

A continuación, describiremos los algoritmos que tenemos que implementar:

- **Add:**

Añade un libro a la base de datos y actualiza el índice. Todo esto, teniendo en cuenta las estrategias de inserción mencionadas anteriormente. A la vez, realiza una búsqueda binaria que comprueba si la clave proporcionada existe. Si no existe, mira en el listado de registros borrados si existe algún hueco capaz de almacenar el libro. A todo esto, actualiza el índice y finalmente, muestra por pantalla un mensaje confirmando que el libro se ha añadido a la base de datos.

Ejemplo:

Input: add 12345|978-2-12345680-3|El Quijote|catedra

Si la clave existe:

Output: Record with BookID=XXXX exists.

Y si no existe:

Output: Record with BookID=12345 has been added to the database

- **Find:**

Realiza una búsqueda binaria en el índice. Si la clave se encuentra, imprime por pantalla el registro encontrado mediante el siguiente mensaje: 12345|978-2-12345680-3|El Quijote|catedra. En el caso de que no se encuentre, imprime por pantalla que el registro no se ha encontrado: Record with bookId=12345 does not exist.

Ejemplo:

Input: find 12345

Si la clave se encuentra:

Output: 12345|978-2-12345680-3|El Quijote|catedra

Si no se encuentra:

Output: Record with bookId=12345 does not exist

- **Del:**

Borra el registro del fichero de datos y actualiza tanto el índice como la lista de registro borrados. Esto se consigue realizando una búsqueda binaria del registro. Si estamos en el caso de que la clave existe, se actualiza el índice y se añade en la lista de registros borrados. Por último, se muestra por pantalla el siguiente mensaje: Record with bookId=12345 has been deleted.

Ejemplo:

Input: del 12345

Si la clave existe:

Output: Record with bookId=12345 has been deleted

Si no existe:

Output: Record with bookId=12345 does not exist

- **Exit:**

Se finaliza el programa pero se debe:

- Cerrar el fichero con los registros de los libros
- Guardar el índice y el listado de registros borrados
- Liberar todos los recursos.

- **printInd:**

Imprime el contenido del índice ordenado de la siguiente manera:

Entry #0

key: #12345

offset: #46

size: #36

Entry #1

key: #12346

offset: #0

size: #38

- **printLst:**

Imprime el contenido del listado de registros borrados de la siguiente manera:

Entry #0

offset: #90

size: #41

Entry #1

offset: #46

size: #36

- **printRec:**

Imprime el fichero con los libros ordenados según el índice e ignorando los registros borrados:

12345|978-2-12345680-3|El Quijote|Catedra

12346|978-2-12345681-3|La busca|Catedra

- **Dinamic Array:**

En este fichero, declararemos varias funciones para su funcionamiento:

initArray:

Inicializa el array del índice

freeArray:

Libera la memoria del array del índice

insertArray:

Inserta un elemento en el array

IndexBinSearch:

Realiza una búsqueda binaria del libro introducido

initArrayDel:

Inicializa el array para los libros borradas

freeArrayDel:

Libera la memoria del array para libros borrados

insertArrayDel:

Inserta un elemento en el array de libros borrados

Respecto a los tests, los comprobamos uno a uno y resultó que nos funcionaban todos ya que se nos mostraban los mensajes que indicaban ese correcto funcionamiento.