Fundamentos de Bases de Datos 2025/2026

Práctica 3

Shaofan Xu y Alejandro Zheng (120/127)

1. Introducción

El objetivo de esta práctica es desarrollar un programa para gestionar una base de datos de una biblioteca, donde almacenamos la información de los libros (como el título, ISBN y editorial) en un fichero binario dentro del disco.

Para que el programa funcione correctamente y mantenga todo organizado, utilizamos dos mecanismos principales en la memoria del ordenador:

- **Un Índice (listado de índice):** Funciona como una tabla de contenidos ordenada. Relaciona el identificador de cada libro con su ubicación exacta dentro del fichero, lo que nos permite saber dónde está guardado cada registro.
- **Una lista de huecos libres (listado de borrados):** Cuando borramos un libro, en lugar de eliminarlo físicamente y mover el resto de los datos, simplemente anotamos en una lista que ese espacio ha quedado libre. De esta forma, cuando queramos añadir un libro nuevo, podemos consultar esta lista y reutilizar esos huecos vacíos siguiendo reglas como elegir el que mejor encaje (Best Fit, ordena de menor a mayor) o el primero que encontremos (First Fit, no ordena) o el más grande que encaje（Worst Fit, ordena de mayor a menor por size）.

Finalmente, para no perder esta organización, tanto el índice como la lista de huecos se guardan en sus propios archivos al cerrar el programa y se recuperan al volver a abrirlo.

2. Estructura del proyecto

Para tener un proyecto más estructurado, hemos decidido a modularizar, es decir, dividir nuestro proyecto en varios módulos, entonces una idea natural de dividirlo es por las funciones más importantes, así que tenemos los siguientes módulos:

- **add (add.c y add.h)**, que implementa todas las funciones necesarias para añadir un libro a la base de datos.
- **del (del.c y del.h)**, que implementa todas las funciones necesarias para borrar un libro de la base de datos.
- **find (find.c y find.h)**, implementa todas las funciones necesarias para buscar un libro del base de datos.

Además, para los macros y los TAD, hemos definido en **types.h.** Lo cual, de esta manera, el proyecto se queda modularizada.

3. Descripción de algoritmo implementado:

- **Add:** se encuentra implementada en **add.c** (función **add_book**), su función es añadir un libro al base de datos, pues, se usa la función **strtok** para determinar cada uno de los campos del libro, y posteriormente, se almacena en un fichero .db donde contendrá todas las informaciones del libro.

También comprueba que el libro que se desee añadir no está en el base de datos, usando **binary_search** implementada en **find.c**. Luego, se comprueba si hay huecos libres en el listado de borrados usando la función **find_and_use_hole** implementada en **del.c**.

Además, para facilitar la búsqueda también guardamos la información (offset y size) del libro como índice en un fichero. ind**,** ya que de esta manera la velocidad de búsqueda es mucha más rápido (en este caso, para los índices primero se guarda en el array dinámico de índice y posteriormente trás el exit se guarda en el fichero).

- **Find**: se encuentra implementada en **find.c** (función **find**), esta función con una clave dada se busca la existencia de un registro. En primer lugar, se hace una búsqueda binaria **(binary_search** implementada también en **find.c)** en el índice. En caso de que existiera recupera el offset y tamaño del registro a través de índice. En segundo lugar, con el offset localiza donde está registro y van recuperando las informaciones de registro y finalmente se imprime por pantalla el registro completo.
- **Del:** se encuentra implementada en **del.c**(función del_book), busca en listado de índice si está el libro que se desee borrar, en el caso de

que no existe, no hará nada salvo imprimir un mensaje; y en el caso afirmativo, se busca la posición del libro que queremos borrar en el listado del índice con **binary_search** (implementada en **find.c**), y posteriormente se guarda en el listado de borrado y se borra el libro del listado de índice. El listado de borrados se guardará en un fichero trás el comando "exit".

- **exit:** en **library.c**, tras ejecutar "exit", el programa termina su ejecución, guardando los listados de borrados y listados de índices en ficheros correspondientes, así liberando todas las memorias que ha reservados.
- **printInd:** implementado en library.c, trás su ejecución en la pantalla muestra todos los índices y sus contenidos en el orden que está en la memoria siguiendo en un formato correcto de primero su orden, el número de key, el offset y el tamaño.
- **printLst:** implementado en library.c, trás su ejecución muestra por la pantalla todos los huecos libres guardado en el listado de borrados, así, el offset y tamaño del hueco, usando el array de listado de borrados.
- **printRec:** implementado en library.c, una vez ejecutado se imprime los libros siguiendo el orden de índice y olvidar de los libros borrado. Para ello, se recupera el offset y tamaño a través de índice. Luego,

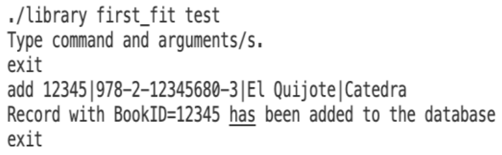
localiza el libro con el offset obtenido, ir recuperando toda la información de libro e imprime en la pantalla. De forma análoga con todos los libros.

4. Ejemplos de algoritmo implementado:

- **Add:**

a. Añadir un libro:

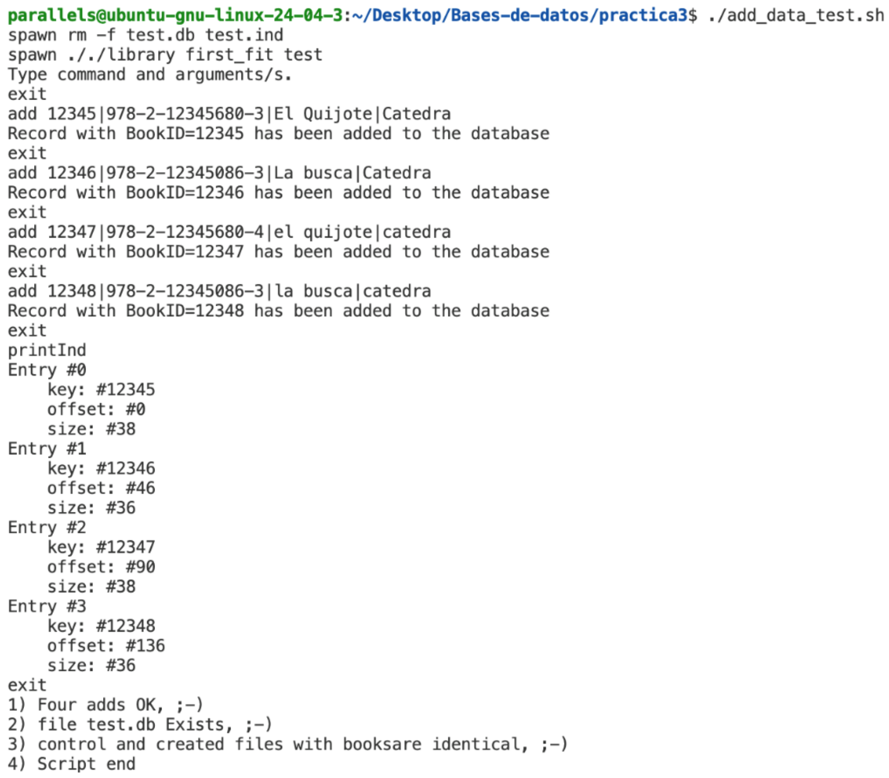
add 12345|978-2-12345680-3|El Quijote|Catedra



```
./library first_fit test
Type command and arguments/s.
exit
add 12345|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12345 has been added to the database
exit
```

Lo cual el libro se ha insertado correctamente en el base de datos.

b. add_data_test.sh



```
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ ./add_data_test.sh
spawn rm -f test.db test.ind
spawn ././library first_fit test
Type command and arguments/s.
exit
add 12345|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12345 has been added to the database
exit
add 12346|978-2-12345086-3|La busca|Catedra
Record with BookID=12346 has been added to the database
exit
add 12347|978-2-12345680-4|el quijote|catedra
Record with BookID=12347 has been added to the database
exit
add 12348|978-2-12345086-3|la busca|catedra
Record with BookID=12348 has been added to the database
exit
printInd
Entry #0
    key: #12345
    offset: #0
    size: #38
Entry #1
    key: #12346
    offset: #46
    size: #36
Entry #2
    key: #12347
    offset: #90
    size: #38
Entry #3
    key: #12348
    offset: #136
    size: #36
exit
1) Four adds OK, ;-)
2) file test.db Exists, ;-)
3) control and created files with booksare identical, ;-)
4) Script end
```

c. Borrar y reutilizar el espacio.

Ejecutando los siguientes comandos en el programa:

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

add 12346|978-2-12345086-3|La busca|Catedra

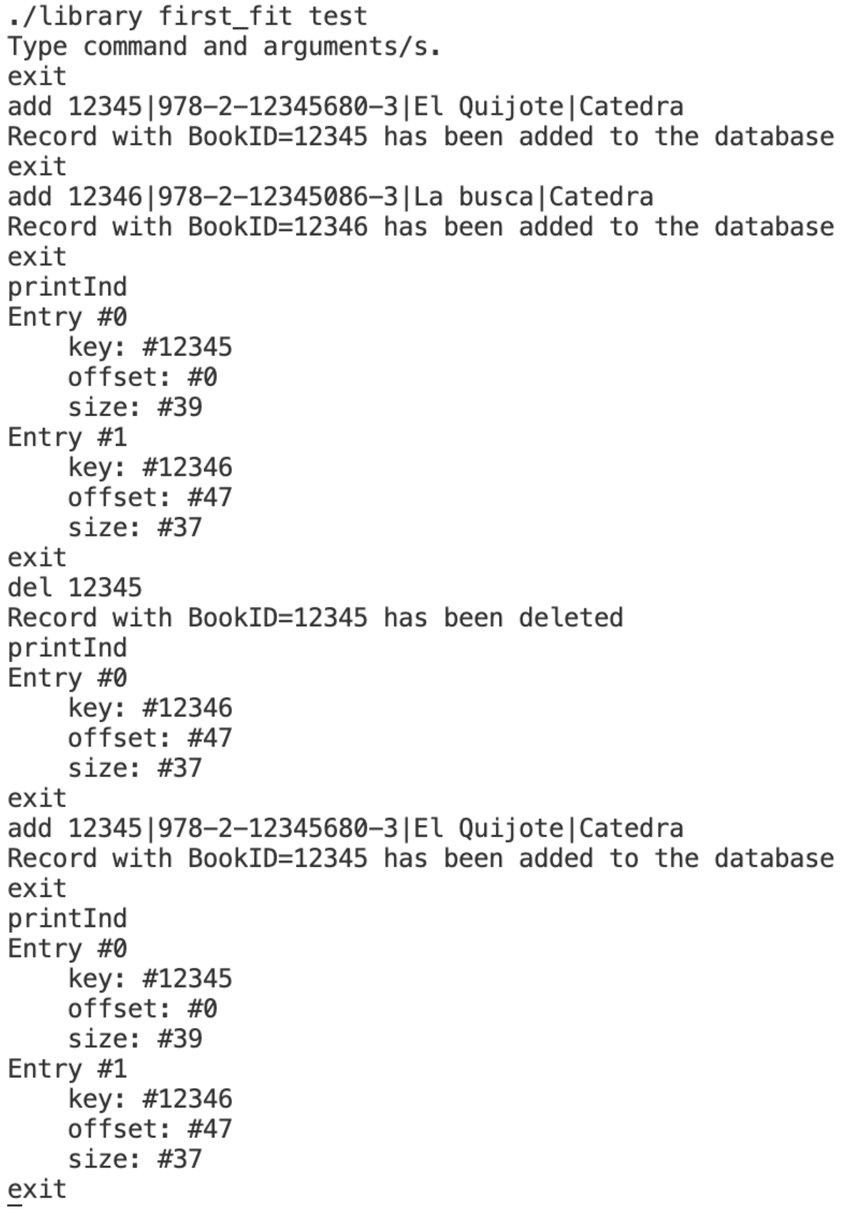
printInd

del 12345

printInd

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

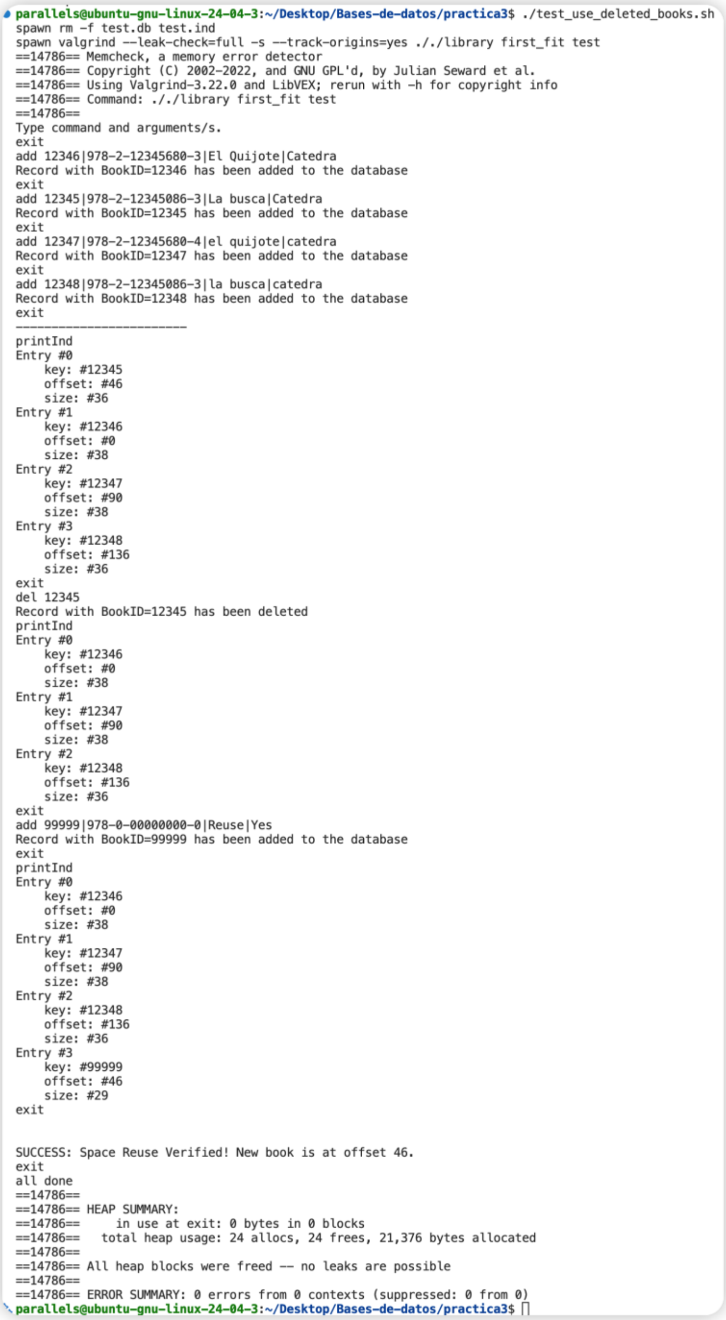
printInd



```
./library first_fit test
Type command and arguments/s.
exit
add 12345|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12345 has been added to the database
exit
add 12346|978-2-12345086-3|La busca|Catedra
Record with BookID=12346 has been added to the database
exit
printInd
Entry #0
    key: #12345
    offset: #0
    size: #39
Entry #1
    key: #12346
    offset: #47
    size: #37
exit
del 12345
Record with BookID=12345 has been deleted
printInd
Entry #0
    key: #12346
    offset: #47
    size: #37
exit
add 12345|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12345 has been added to the database
exit
printInd
Entry #0
    key: #12345
    offset: #0
    size: #39
Entry #1
    key: #12346
    offset: #47
    size: #37
exit
```

Podemos observar que, hemos añadido 2 libros, y después borramos el primer libro, y comprobamos que se ha borrado correctamente, luego reinsertamos el primer libro, lo cual podemos ver que se ha reutilizado el espacio con el mismo offset que antes.

d. **Además, probamos el script que hemos diseñado para reutilizar el espacio test_use_deleted_books.sh**



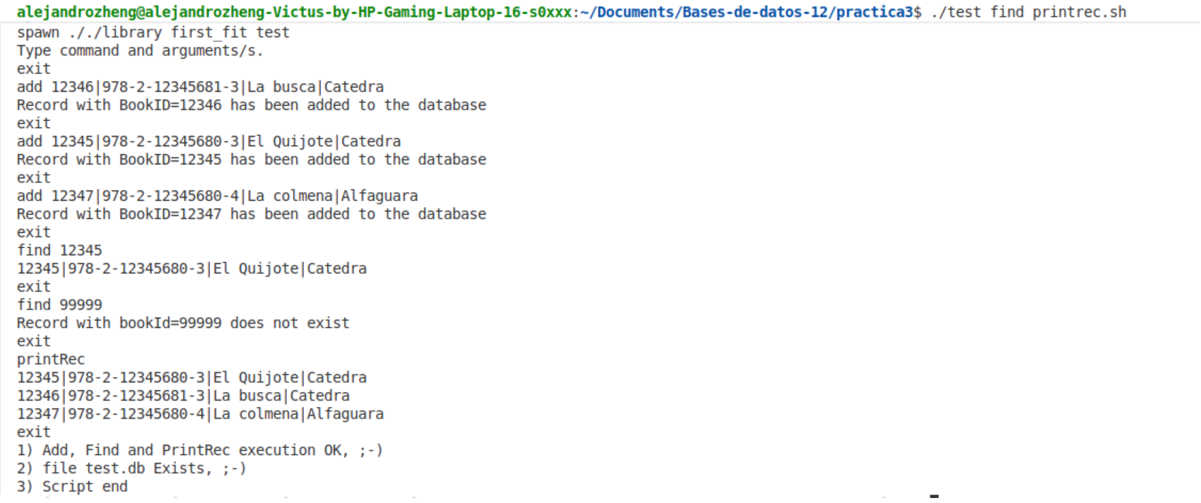
```
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ ./test_use_deleted_books.sh
spawn rm -f test.db test.ind
spawn valgrind --leak-check=full -s --track-origins=yes ././library first_fit test
==14786== Memcheck, a memory error detector
==14786== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==14786== Using Valgrind-3.22.0 and LibVEX; rerun with -h for copyright info
==14786== Command: ././library first_fit test
==14786==
Type command and arguments/s.
exit
add 12346|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12346 has been added to the database
exit
add 12345|978-2-12345086-3|La busca|Catedra
Record with BookID=12345 has been added to the database
exit
add 12347|978-2-12345680-4|el quijote|catedra
Record with BookID=12347 has been added to the database
exit
add 12348|978-2-12345086-3|la busca|catedra
Record with BookID=12348 has been added to the database
exit
------------------------
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12347
    offset: #90
    size: #38
Entry #3
    key: #12348
    offset: #136
    size: #36
exit
del 12345
Record with BookID=12345 has been deleted
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
Entry #1
    key: #12347
    offset: #90
    size: #38
Entry #2
    key: #12348
    offset: #136
    size: #36
exit
add 99999|978-0-00000000-0|Reuse|Yes
Record with BookID=99999 has been added to the database
exit
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
Entry #1
    key: #12347
    offset: #90
    size: #38
Entry #2
    key: #12348
    offset: #136
    size: #36
Entry #3
    key: #99999
    offset: #46
    size: #29
exit


SUCCESS: Space Reuse Verified! New book is at offset 46.
exit
all done
==14786==
==14786== HEAP SUMMARY:
==14786==     in use at exit: 0 bytes in 0 blocks
==14786==   total heap usage: 24 allocs, 24 frees, 21,376 bytes allocated
==14786==
==14786== All heap blocks were freed -- no leaks are possible
==14786==
==14786== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ 
```

Similar que el caso anterior, en este caso, añadimos 3 libros, y borramos, el primer libro con size=36 y offset=46. Ahora insertamos un nuevo libro con menor size que 36 y book_id=99999, lo cual, el resultado esperado es que al insertarlo tenga offset=46. Y, en definitiva, el offset el nuevo libro es 46.

- **Find**:

Esta función está localizada en el fichero llamado **find.c** y se puede comprobar su funcionamiento con el script **test_find_printrec.sh.** Donde se hace dos find, uno con un libro que ha sido previamente añadido y otro find con un libro que no ha sido añadió. De esta forma podemos comprobar la salida esperada de esta función.

La siguiente imagen muestra el resultado de ejecución de dicho script:



```
alejandrozheng@alejandrozheng-Victus-by-HP-Gaming-Laptop-16-s0xxx:~/Documents/Bases-de-datos-12/practica3$ ./test find printrec.sh
spawn ././library first_fit test
Type command and arguments/s.
exit
add 12346|978-2-12345681-3|La busca|Catedra
Record with BookID=12346 has been added to the database
exit
add 12345|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12345 has been added to the database
exit
add 12347|978-2-12345680-4|La colmena|Alfaguara
Record with BookID=12347 has been added to the database
exit
find 12345
12345|978-2-12345680-3|El Quijote|Catedra
exit
find 99999
Record with bookId=99999 does not exist
exit
printRec
12345|978-2-12345680-3|El Quijote|Catedra
12346|978-2-12345681-3|La busca|Catedra
12347|978-2-12345680-4|La colmena|Alfaguara
exit
1) Add, Find and PrintRec execution OK, ;-)
2) file test.db Exists, ;-)
3) Script end
```

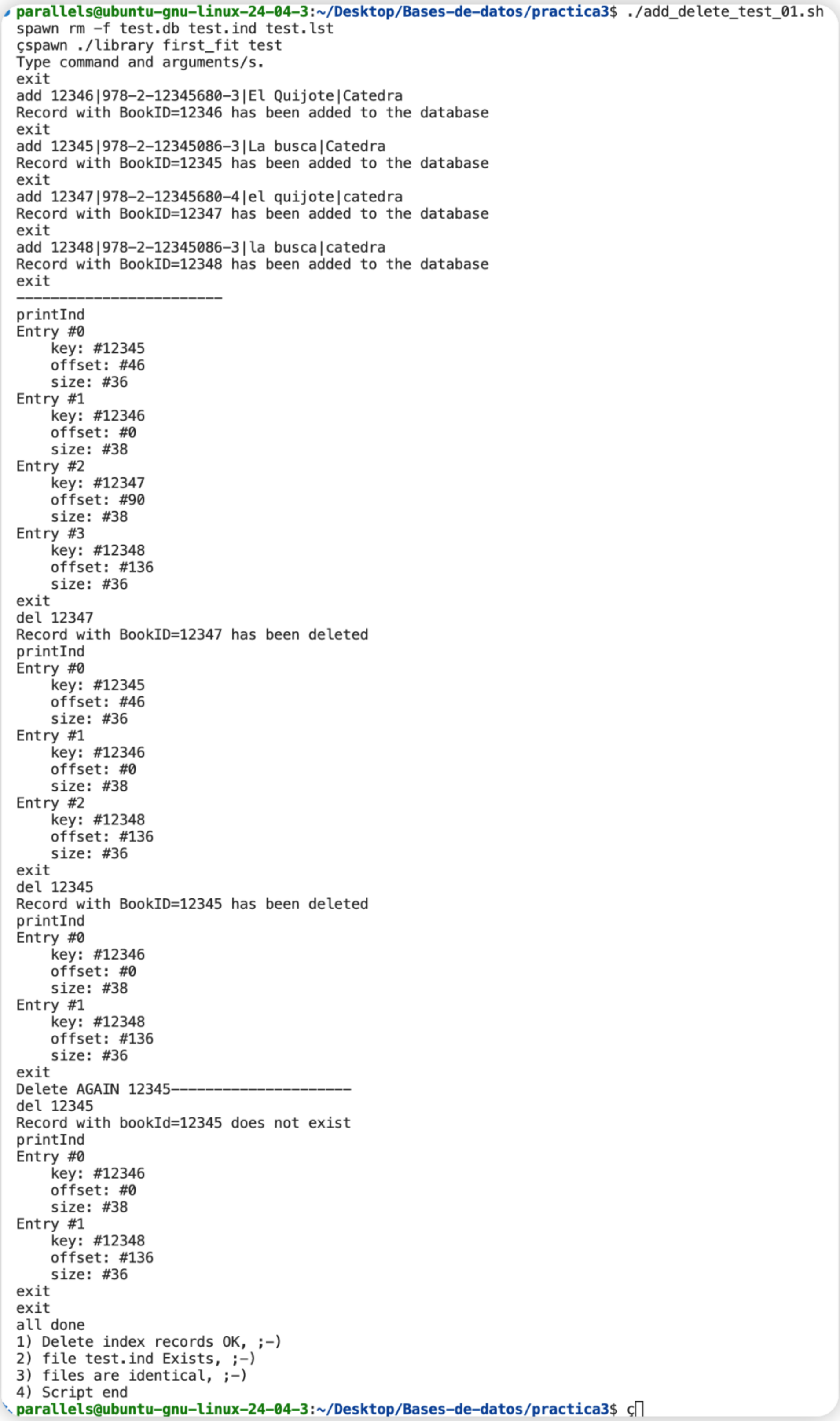
Donde podemos observar que añade tres libros con book_id:

12346,12345 y 12347. Luego hacer un find de libro con book_id=12345, el resultado que obtiene es la información completa de ese libro. Sin embargo, cuando hacemos un find de book_id=99999, como ese libro no ha sido añadido previamente el resultado que muestra en la pantalla es “Record with bookId=99999 does not exist”. De esta forma podemos comprobar el funcionamiento de esta función es correcto.

- **Del:**

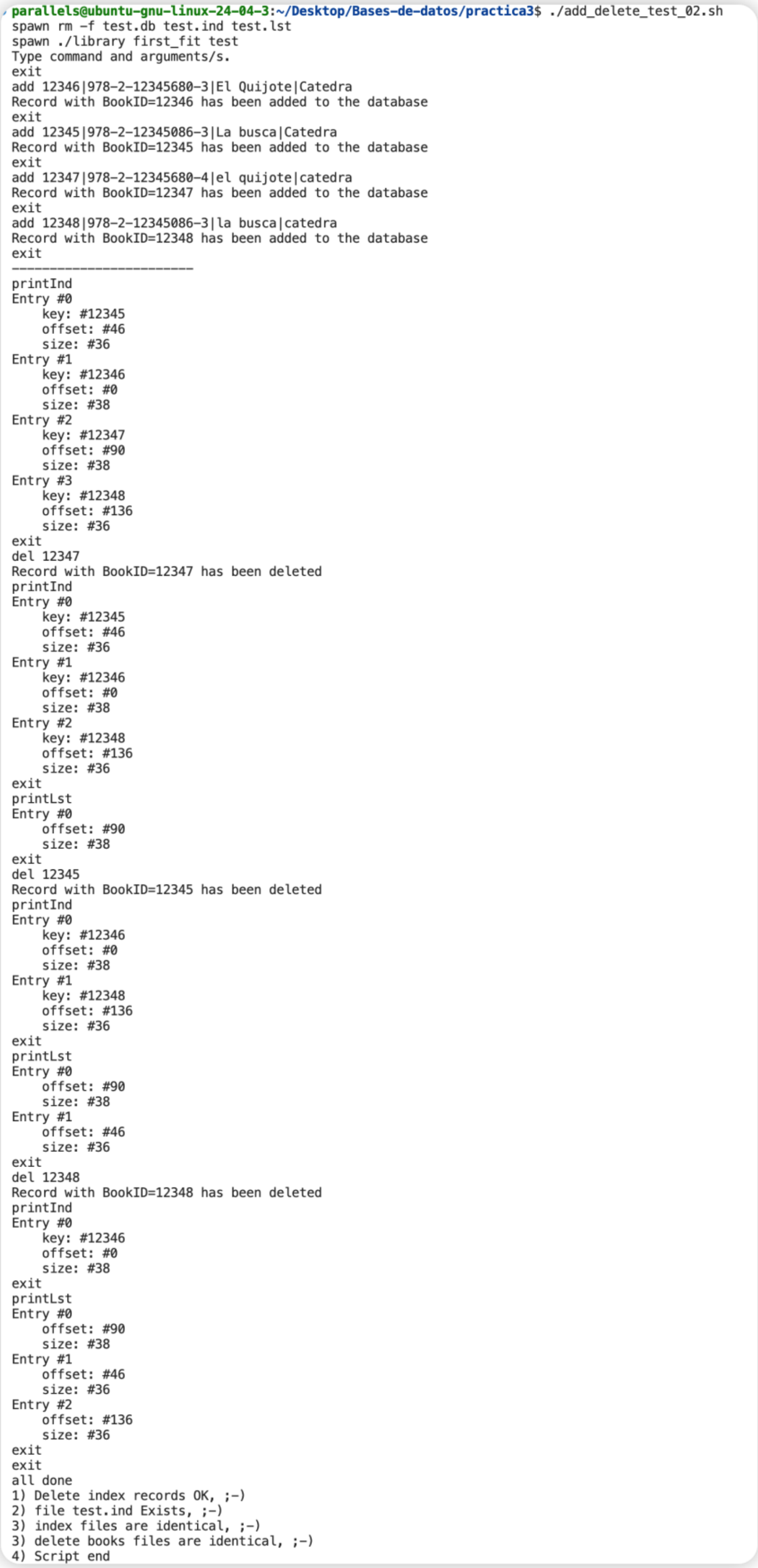
Para probar el funcionamiento de borrar un libro nos basta con probar las siguientes scripts proporcionadas:

a. add_delete_test_01.sh



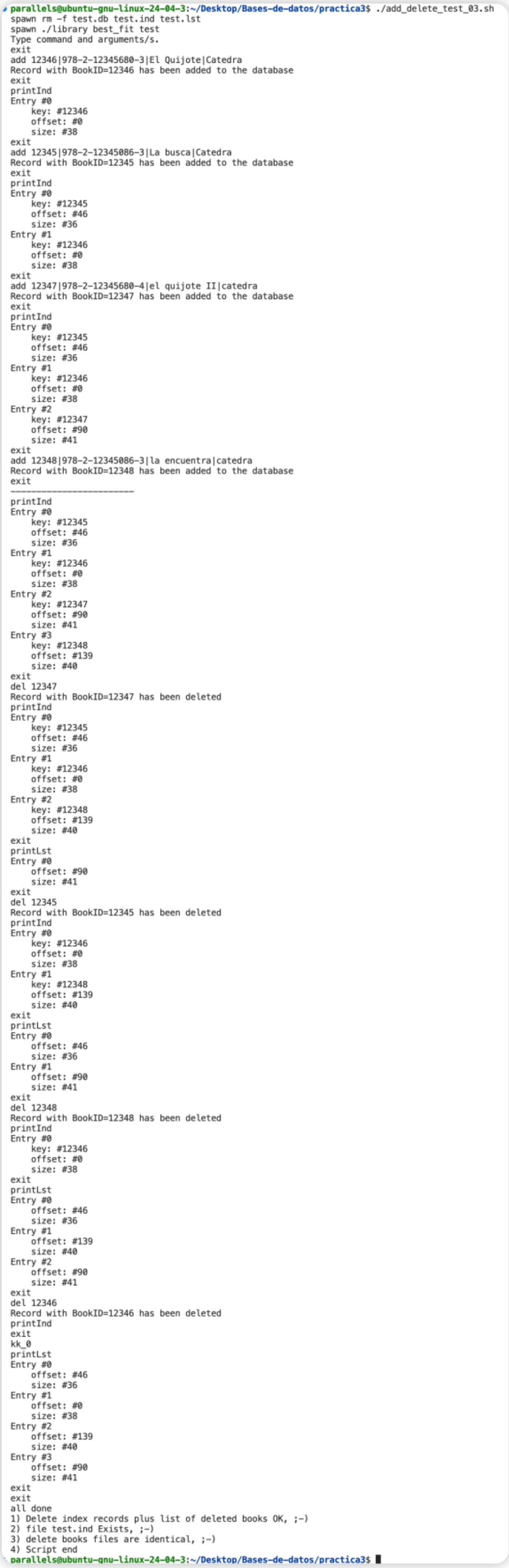
```
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ ./add_delete_test_01.sh
spawn rm -f test.db test.ind test.lst
çspawn ./library first_fit test
Type command and arguments/s.
exit
add 12346|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12346 has been added to the database
exit
add 12345|978-2-12345086-3|La busca|Catedra
Record with BookID=12345 has been added to the database
exit
add 12347|978-2-12345680-4|el quijote|catedra
Record with BookID=12347 has been added to the database
exit
add 12348|978-2-12345086-3|la busca|catedra
Record with BookID=12348 has been added to the database
exit
------------------------
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12347
    offset: #90
    size: #38
Entry #3
    key: #12348
    offset: #136
    size: #36
exit
del 12347
Record with BookID=12347 has been deleted
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12348
    offset: #136
    size: #36
exit
del 12345
Record with BookID=12345 has been deleted
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
Entry #1
    key: #12348
    offset: #136
    size: #36
exit
Delete AGAIN 12345---------------------
del 12345
Record with bookId=12345 does not exist
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
Entry #1
    key: #12348
    offset: #136
    size: #36
exit
exit
all done
1) Delete index records OK, ;-)
2) file test.ind Exists, ;-)
3) files are identical, ;-)
4) Script end
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ ç
```

b. add_delete_test_02.sh



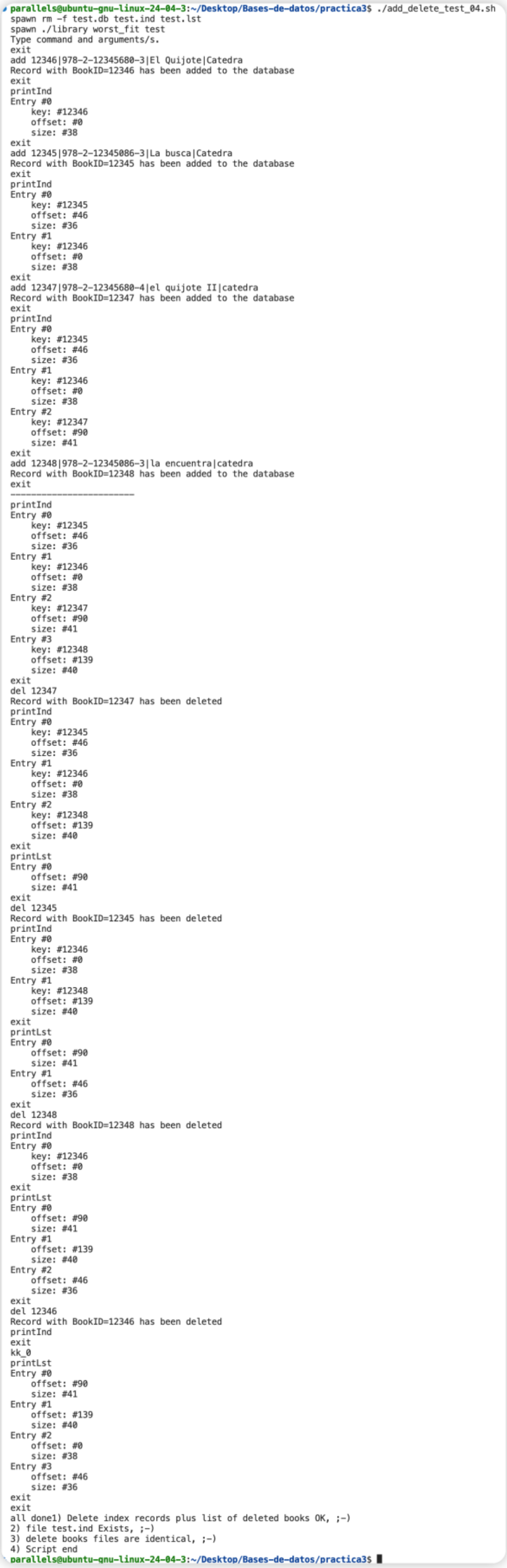
```
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ ./add_delete_test_02.sh
spawn rm -f test.db test.ind test.lst
spawn ./library first_fit test
Type command and arguments/s.
exit
add 12346|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12346 has been added to the database
exit
add 12345|978-2-12345086-3|La busca|Catedra
Record with BookID=12345 has been added to the database
exit
add 12347|978-2-12345680-4|el quijote|catedra
Record with BookID=12347 has been added to the database
exit
add 12348|978-2-12345086-3|la busca|catedra
Record with BookID=12348 has been added to the database
exit
------------------------
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12347
    offset: #90
    size: #38
Entry #3
    key: #12348
    offset: #136
    size: #36
exit
del 12347
Record with BookID=12347 has been deleted
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12348
    offset: #136
    size: #36
exit
printLst
Entry #0
    offset: #90
    size: #38
exit
del 12345
Record with BookID=12345 has been deleted
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
Entry #1
    key: #12348
    offset: #136
    size: #36
exit
printLst
Entry #0
    offset: #90
    size: #38
Entry #1
    offset: #46
    size: #36
exit
del 12348
Record with BookID=12348 has been deleted
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
exit
printLst
Entry #0
    offset: #90
    size: #38
Entry #1
    offset: #46
    size: #36
Entry #2
    offset: #136
    size: #36
exit
exit
all done
1) Delete index records OK, ;-)
2) file test.ind Exists, ;-)
3) index files are identical, ;-)
3) delete books files are identical, ;-)
4) Script end
```

c. add_delete_test_03.sh



```
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ ./add_delete_test_03.sh
spawn rm -f test.db test.ind test.lst
spawn ./library best_fit test
Type command and arguments/s.
exit
add 12346|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12346 has been added to the database
exit
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
exit
add 12345|978-2-12345086-3|La busca|Catedra
Record with BookID=12345 has been added to the database
exit
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
exit
add 12347|978-2-12345680-4|el quijote II|catedra
Record with BookID=12347 has been added to the database
exit
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12347
    offset: #90
    size: #41
exit
add 12348|978-2-12345086-3|la encuentra|catedra
Record with BookID=12348 has been added to the database
exit
------------------------
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12347
    offset: #90
    size: #41
Entry #3
    key: #12348
    offset: #139
    size: #40
exit
del 12347
Record with BookID=12347 has been deleted
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12348
    offset: #139
    size: #40
exit
printLst
Entry #0
    offset: #90
    size: #41
exit
del 12345
Record with BookID=12345 has been deleted
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
Entry #1
    key: #12348
    offset: #139
    size: #40
exit
printLst
Entry #0
    offset: #46
    size: #36
Entry #1
    offset: #90
    size: #41
exit
del 12348
Record with BookID=12348 has been deleted
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
exit
printLst
Entry #0
    offset: #46
    size: #36
Entry #1
    offset: #139
    size: #40
Entry #2
    offset: #90
    size: #41
exit
del 12346
Record with BookID=12346 has been deleted
printInd
exit
kk_0
printLst
Entry #0
    offset: #46
    size: #36
Entry #1
    offset: #0
    size: #38
Entry #2
    offset: #139
    size: #40
Entry #3
    offset: #90
    size: #41
exit
exit
all done
1) Delete index records plus list of deleted books OK, ;-)
2) file test.ind Exists, ;-)
3) delete books files are identical, ;-)
4) Script end
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ ▌
```

d. add_delete_test_04.sh



```
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ ./add_delete_test_04.sh
spawn rm -f test.db test.ind test.lst
spawn ./library worst_fit test
Type command and arguments/s.
exit
add 12346|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12346 has been added to the database
exit
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
exit
add 12345|978-2-12345086-3|La busca|Catedra
Record with BookID=12345 has been added to the database
exit
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
exit
add 12347|978-2-12345680-4|el quijote II|catedra
Record with BookID=12347 has been added to the database
exit
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12347
    offset: #90
    size: #41
exit
add 12348|978-2-12345086-3|la encuentra|catedra
Record with BookID=12348 has been added to the database
exit
------------------------
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12347
    offset: #90
    size: #41
Entry #3
    key: #12348
    offset: #139
    size: #40
exit
del 12347
Record with BookID=12347 has been deleted
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12348
    offset: #139
    size: #40
exit
printLst
Entry #0
    offset: #90
    size: #41
exit
del 12345
Record with BookID=12345 has been deleted
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
Entry #1
    key: #12348
    offset: #139
    size: #40
exit
printLst
Entry #0
    offset: #90
    size: #41
Entry #1
    offset: #46
    size: #36
exit
del 12348
Record with BookID=12348 has been deleted
printInd
Entry #0
    key: #12346
    offset: #0
    size: #38
exit
printLst
Entry #0
    offset: #90
    size: #41
Entry #1
    offset: #139
    size: #40
Entry #2
    offset: #46
    size: #36
exit
del 12346
Record with BookID=12346 has been deleted
printInd
exit
kk_0
printLst
Entry #0
    offset: #90
    size: #41
Entry #1
    offset: #139
    size: #40
Entry #2
    offset: #0
    size: #38
Entry #3
    offset: #46
    size: #36
exit
exit
all done1) Delete index records plus list of deleted books OK, ;-)
2) file test.ind Exists, ;-)
3) delete books files are identical, ;-)
4) Script end
parallels@ubuntu-gnu-linux-24-04-3:~/Desktop/Bases-de-datos/practica3$ 
```

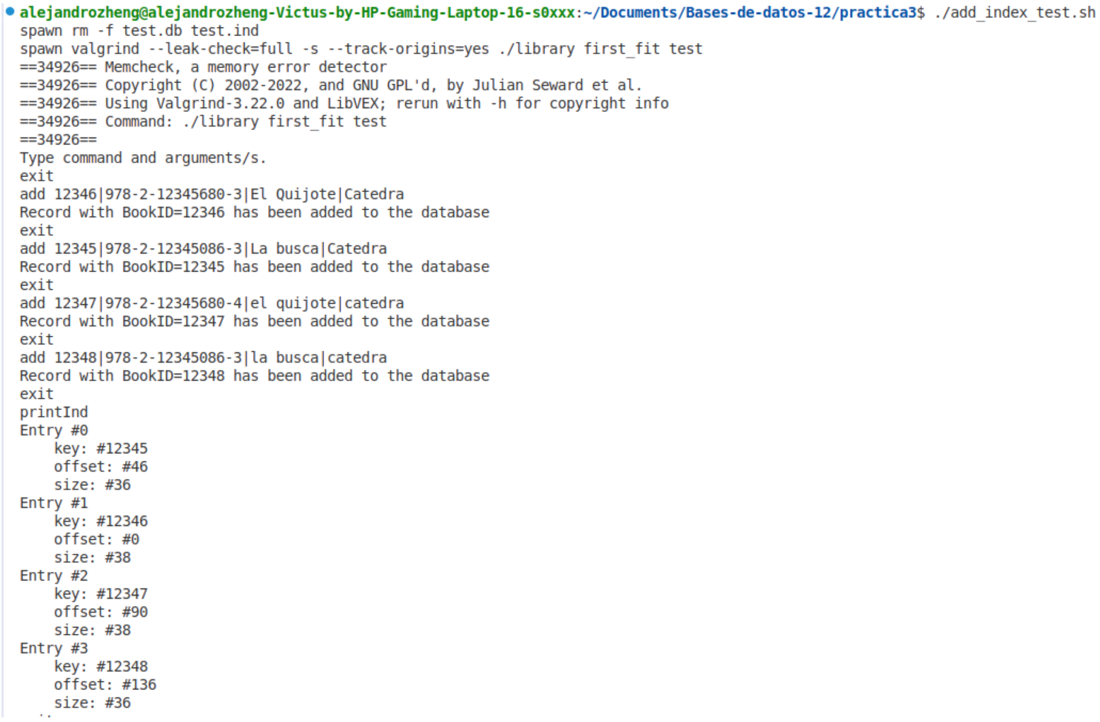
- **Exit:**

  Esta función como anteriormente ha hablado que está en library.c. Como es una función estamos usando continuamente en todos los scripts como el script **test_use_deleted_books.sh, add_index_test.sh** y otros scripts más. Damos por hecho que de esta funcionando perfectamente y no necesitamos hacer un ejemplo específico para “exit”.

- **printInd:**

  Esta función esta implementada en el fichero library.c que imprime los contenidos de índices en la pantalla y podemos comprobar el funcionamiento de esta función con los scripts add_index_test.sh y otros muchos test como add_data_test.sh, add_delete_test_01.sh

  La siguiente imagen muestra la ejecución de script add_index_test.sh:



```
● alejandrozheng@alejandrozheng-Victus-by-HP-Gaming-Laptop-16-s0xxx:~/Documents/Bases-de-datos-12/practica3$ ./add_index_test.sh
  spawn rm -f test.db test.ind
  spawn valgrind --leak-check=full -s --track-origins=yes ./library first_fit test
  ==34926== Memcheck, a memory error detector
  ==34926== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
  ==34926== Using Valgrind-3.22.0 and LibVEX; rerun with -h for copyright info
  ==34926== Command: ./library first_fit test
  ==34926==
  Type command and arguments/s.
  exit
  add 12346|978-2-12345680-3|El Quijote|Catedra
  Record with BookID=12346 has been added to the database
  exit
  add 12345|978-2-12345086-3|La busca|Catedra
  Record with BookID=12345 has been added to the database
  exit
  add 12347|978-2-12345680-4|el quijote|catedra
  Record with BookID=12347 has been added to the database
  exit
  add 12348|978-2-12345086-3|la busca|catedra
  Record with BookID=12348 has been added to the database
  exit
  printInd
  Entry #0
      key: #12345
      offset: #46
      size: #36
  Entry #1
      key: #12346
      offset: #0
      size: #38
  Entry #2
      key: #12347
      offset: #90
      size: #38
  Entry #3
      key: #12348
      offset: #136
      size: #36
```

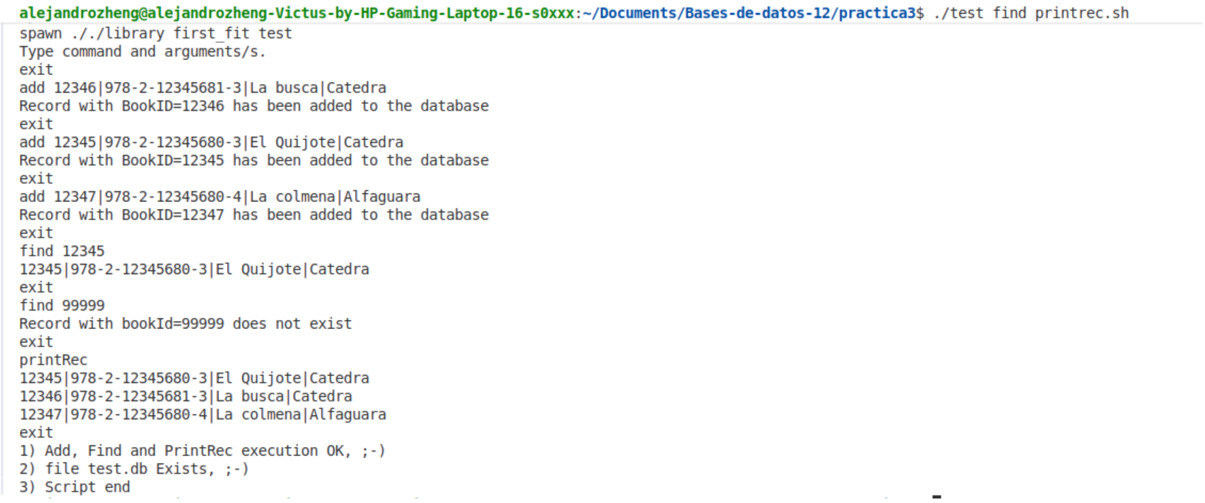
Donde podemos ver que después de insertar ciertos libros, cuando hacemos printInd se muestra los Entry, la clave que es el book_id de los libros luego su offset y su tamaño. Asimismo, comprobamos su funcionamiento correcto.

- **printLst:**

  PrintLst está localizada en el fichero library.c que imprime el contenido del listado de borrado. Este comando es ejecutado varias veces en los tests de prueba: add_delete_test_01.sh, add_delete_test_02.sh, add_delete_test_03.sh y etc. Lo cual, basta ver las imágenes de ejemplos de **Del.**

- **printRec:**

  PrintRec está localizada en el fichero library.c que imprime el contenido de los libros de fichero en el orden de índice guardado. Esta función podemos comprobar en el script llamado **test_find_printrec.sh**



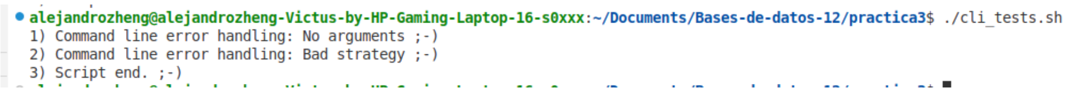
```
alejandrozheng@alejandrozheng-Victus-by-HP-Gaming-Laptop-16-s0xxx:~/Documents/Bases-de-datos-12/practica3$ ./test find printrec.sh
spawn ././library first_fit test
Type command and arguments/s.
exit
add 12346|978-2-12345681-3|La busca|Catedra
Record with BookID=12346 has been added to the database
exit
add 12345|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12345 has been added to the database
exit
add 12347|978-2-12345680-4|La colmena|Alfaguara
Record with BookID=12347 has been added to the database
exit
find 12345
12345|978-2-12345680-3|El Quijote|Catedra
exit
find 99999
Record with bookId=99999 does not exist
exit
printRec
12345|978-2-12345680-3|El Quijote|Catedra
12346|978-2-12345681-3|La busca|Catedra
12347|978-2-12345680-4|La colmena|Alfaguara
exit
1) Add, Find and PrintRec execution OK, ;-)
2) file test.db Exists, ;-)
3) Script end
```

Observando el resultado de script tras ejecución vemos que imprime los contenidos de libros el formato que pide en el enunciado, por lo tanto, su funcionamiento es correcto.

5. El resultado de las pruebas

En este apartado vamos a mostrar los resultados de las pruebas tras su ejecución, algunos test ya han sido mostrado previamente para la explicación de las funciones que hemos implementado según

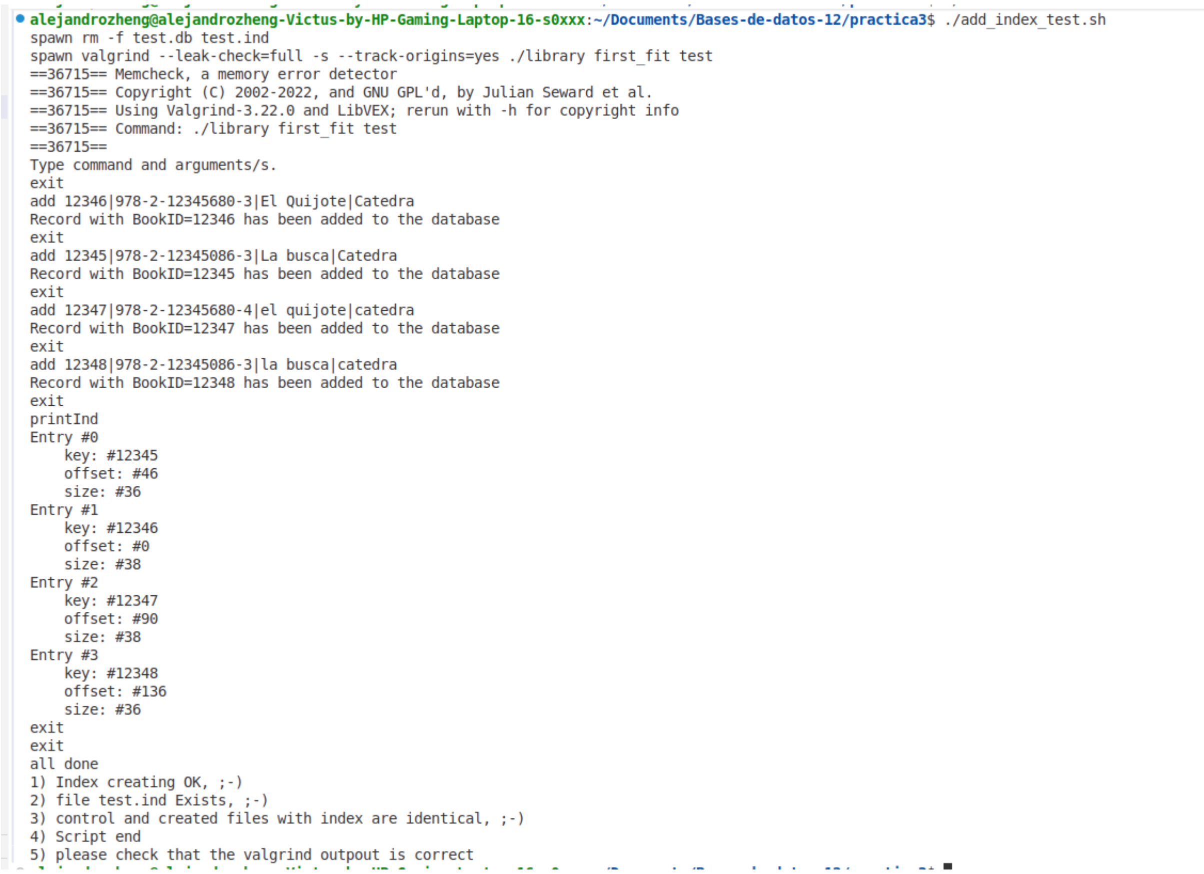
- cli_tests.sh



```
● alejandrozheng@alejandrozheng-Victus-by-HP-Gaming-Laptop-16-s0xxx:~/Documents/Bases-de-datos-12/practica3$ ./cli_tests.sh
  1) Command line error handling: No arguments ;-)
  2) Command line error handling: Bad strategy ;-)
  3) Script end. ;-)
```

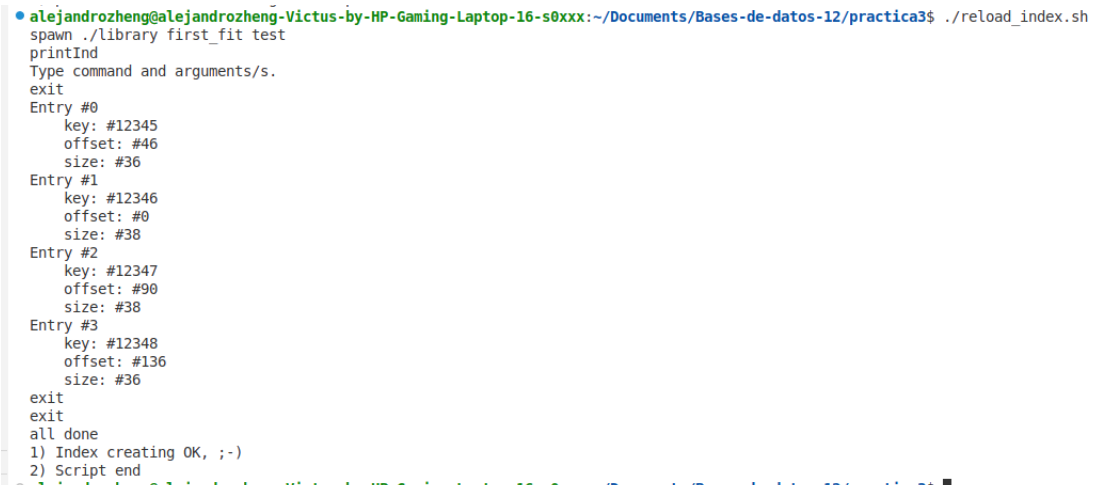
- add_data_test.sh: Ha sido demostrado ya en el ejemplo de **add**.

- add_index_test.sh



```
alejandrozheng@alejandrozheng-Victus-by-HP-Gaming-Laptop-16-s0xxx:~/Documents/Bases-de-datos-12/practica3$ ./add_index_test.sh
spawn rm -f test.db test.ind
spawn valgrind --leak-check=full -s --track-origins=yes ./library first_fit test
==36715== Memcheck, a memory error detector
==36715== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==36715== Using Valgrind-3.22.0 and LibVEX; rerun with -h for copyright info
==36715== Command: ./library first_fit test
==36715==
Type command and arguments/s.
exit
add 12346|978-2-12345680-3|El Quijote|Catedra
Record with BookID=12346 has been added to the database
exit
add 12345|978-2-12345086-3|La busca|Catedra
Record with BookID=12345 has been added to the database
exit
add 12347|978-2-12345680-4|el quijote|catedra
Record with BookID=12347 has been added to the database
exit
add 12348|978-2-12345086-3|la busca|catedra
Record with BookID=12348 has been added to the database
exit
printInd
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12347
    offset: #90
    size: #38
Entry #3
    key: #12348
    offset: #136
    size: #36
exit
exit
all done
1) Index creating OK, ;-)
2) file test.ind Exists, ;-)
3) control and created files with index are identical, ;-)
4) Script end
5) please check that the valgrind outpout is correct
```

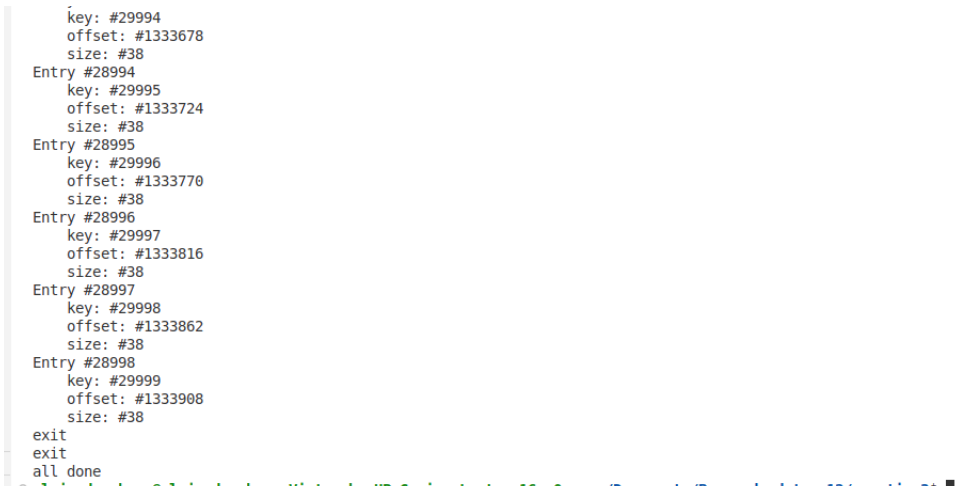
- reload_index.sh



```
alejandrozheng@alejandrozheng-Victus-by-HP-Gaming-Laptop-16-s0xxx:~/Documents/Bases-de-datos-12/practica3$ ./reload_index.sh
spawn ./library first_fit test
printInd
Type command and arguments/s.
exit
Entry #0
    key: #12345
    offset: #46
    size: #36
Entry #1
    key: #12346
    offset: #0
    size: #38
Entry #2
    key: #12347
    offset: #90
    size: #38
Entry #3
    key: #12348
    offset: #136
    size: #36
exit
exit
all done
1) Index creating OK, ;-)
2) Script end
```

- many_entries.sh

Como el output de este script test es muy largo solo mostramos los resultados finales



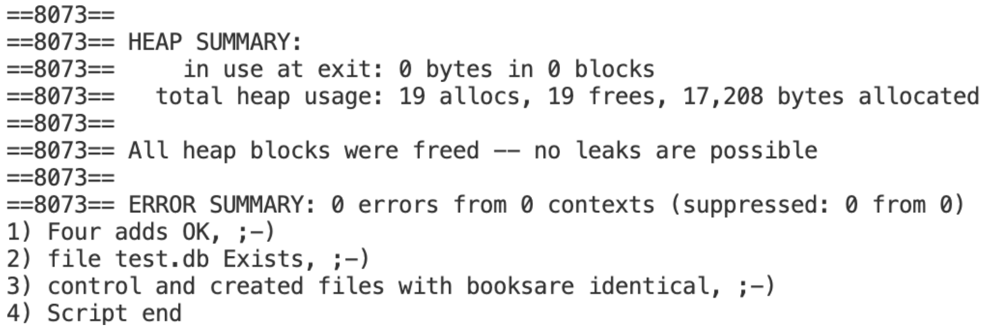
```
    key: #29994
    offset: #1333678
    size: #38
Entry #28994
    key: #29995
    offset: #1333724
    size: #38
Entry #28995
    key: #29996
    offset: #1333770
    size: #38
Entry #28996
    key: #29997
    offset: #1333816
    size: #38
Entry #28997
    key: #29998
    offset: #1333862
    size: #38
Entry #28998
    key: #29999
    offset: #1333908
    size: #38
exit
exit
all done
```

- Los scripts de pruebas de delete como ya han sido mostrado en la explicación de funcionamiento de “del”, en este apartado no lo mostramos otra vez para evitar repeticiones de información.

6. Comprobación de valgrind

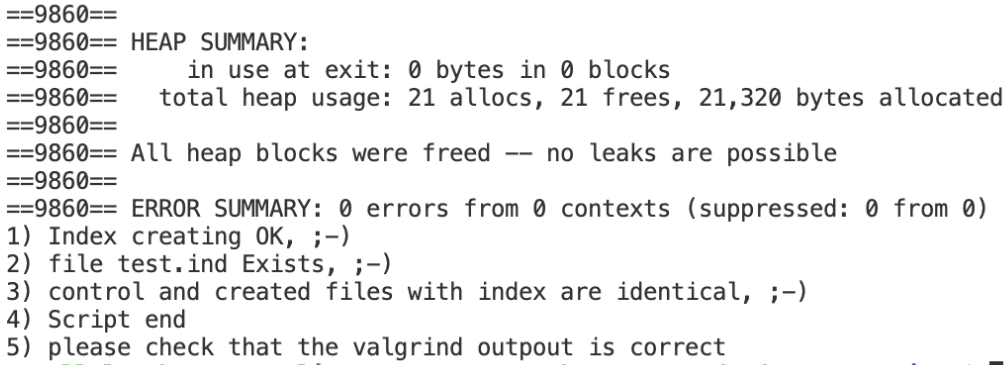
Vamos a ver la salida de valgrind ejecutando las pruebas （*.sh） con valgrind (lo cual, hemos modificado las pruebas para que se ejecuta con valgrind) para comprobar que no hay fugas de memorias ni errores en el programa:

- add_data_test.sh



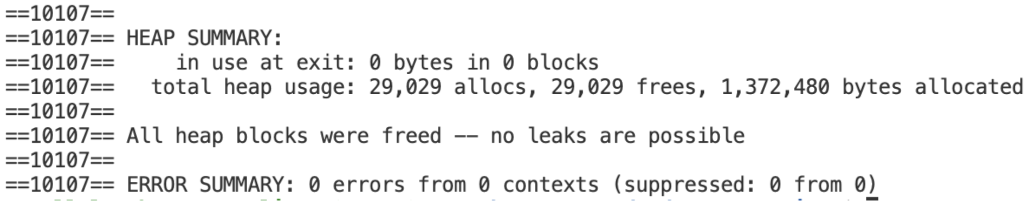
```
==8073==
==8073== HEAP SUMMARY:
==8073==     in use at exit: 0 bytes in 0 blocks
==8073==   total heap usage: 19 allocs, 19 frees, 17,208 bytes allocated
==8073==
==8073== All heap blocks were freed -- no leaks are possible
==8073==
==8073== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
1) Four adds OK, ;-)
2) file test.db Exists, ;-)
3) control and created files with booksare identical, ;-)
4) Script end
```

- add_index_test.sh



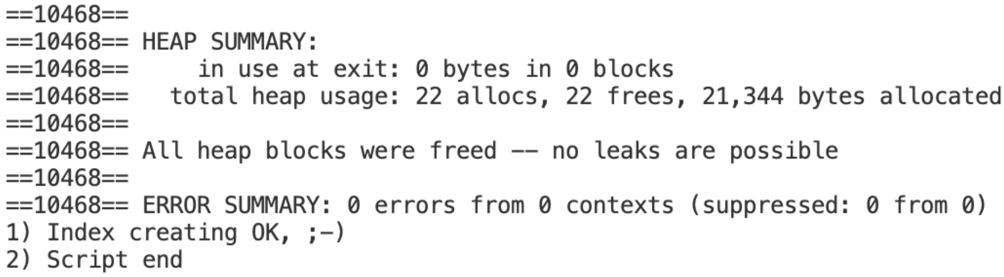
```
==9860==
==9860== HEAP SUMMARY:
==9860==     in use at exit: 0 bytes in 0 blocks
==9860==   total heap usage: 21 allocs, 21 frees, 21,320 bytes allocated
==9860==
==9860== All heap blocks were freed -- no leaks are possible
==9860==
==9860== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
1) Index creating OK, ;-)
2) file test.ind Exists, ;-)
3) control and created files with index are identical, ;-)
4) Script end
5) please check that the valgrind outpout is correct
```

- many_entries.sh



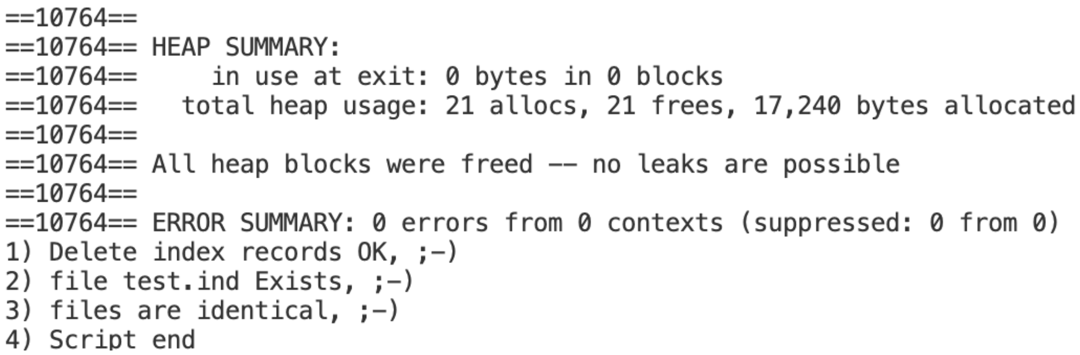
```
==10107==
==10107== HEAP SUMMARY:
==10107==     in use at exit: 0 bytes in 0 blocks
==10107==   total heap usage: 29,029 allocs, 29,029 frees, 1,372,480 bytes allocated
==10107==
==10107== All heap blocks were freed -- no leaks are possible
==10107==
==10107== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

- reload_index.sh



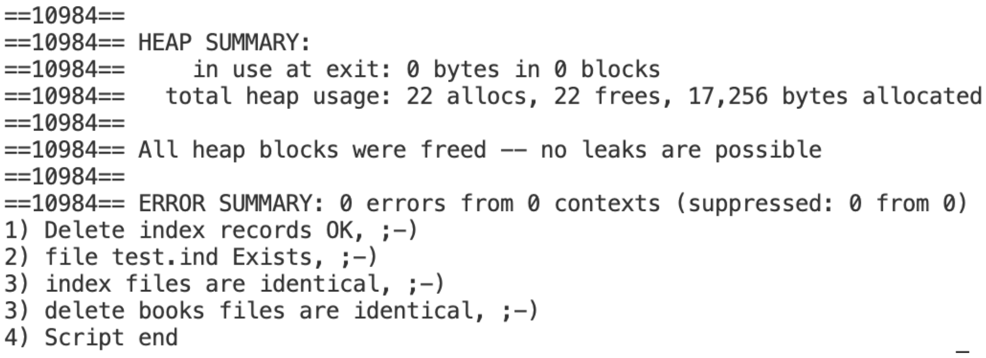
```
==10468==
==10468== HEAP SUMMARY:
==10468==     in use at exit: 0 bytes in 0 blocks
==10468==   total heap usage: 22 allocs, 22 frees, 21,344 bytes allocated
==10468==
==10468== All heap blocks were freed -- no leaks are possible
==10468==
==10468== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
1) Index creating OK, ;-)
2) Script end
```

- add_delete_test_01.sh



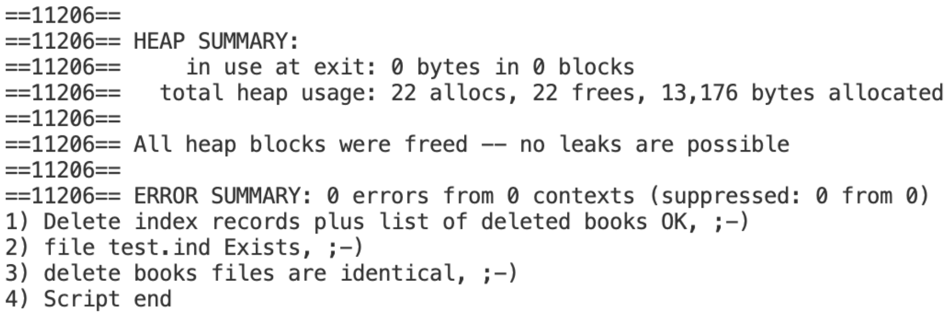
```
==10764==
==10764== HEAP SUMMARY:
==10764==     in use at exit: 0 bytes in 0 blocks
==10764==   total heap usage: 21 allocs, 21 frees, 17,240 bytes allocated
==10764==
==10764== All heap blocks were freed -- no leaks are possible
==10764==
==10764== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
1) Delete index records OK, ;-)
2) file test.ind Exists, ;-)
3) files are identical, ;-)
4) Script end
```

- add_delete_test_02.sh



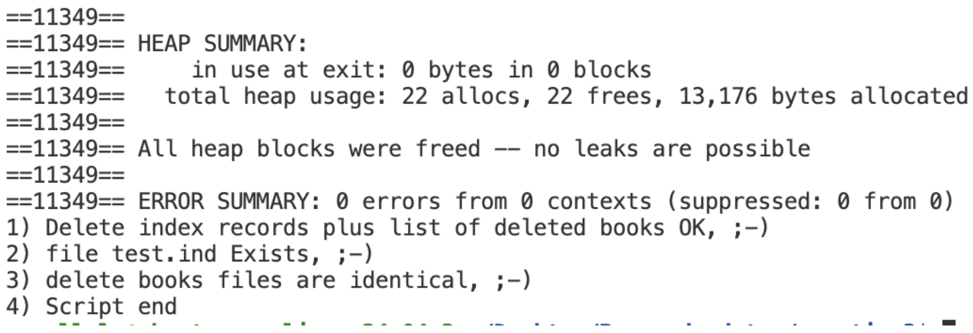
```
==10984==
==10984== HEAP SUMMARY:
==10984==     in use at exit: 0 bytes in 0 blocks
==10984==   total heap usage: 22 allocs, 22 frees, 17,256 bytes allocated
==10984==
==10984== All heap blocks were freed -- no leaks are possible
==10984==
==10984== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
1) Delete index records OK, ;-)
2) file test.ind Exists, ;-)
3) index files are identical, ;-)
3) delete books files are identical, ;-)
4) Script end
```

- add_delete_test_03.sh



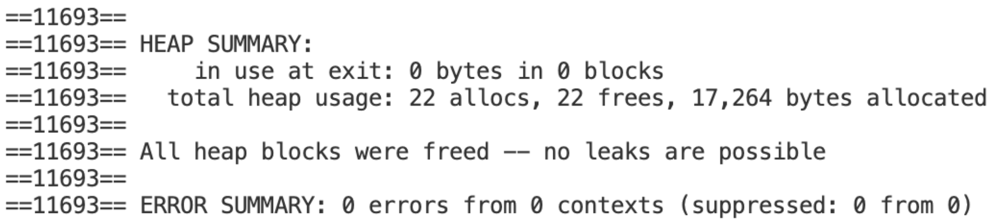
```
==11206==
==11206== HEAP SUMMARY:
==11206==     in use at exit: 0 bytes in 0 blocks
==11206==   total heap usage: 22 allocs, 22 frees, 13,176 bytes allocated
==11206==
==11206== All heap blocks were freed -- no leaks are possible
==11206==
==11206== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
1) Delete index records plus list of deleted books OK, ;-)
2) file test.ind Exists, ;-)
3) delete books files are identical, ;-)
4) Script end
```

- add_delete_test_04.sh



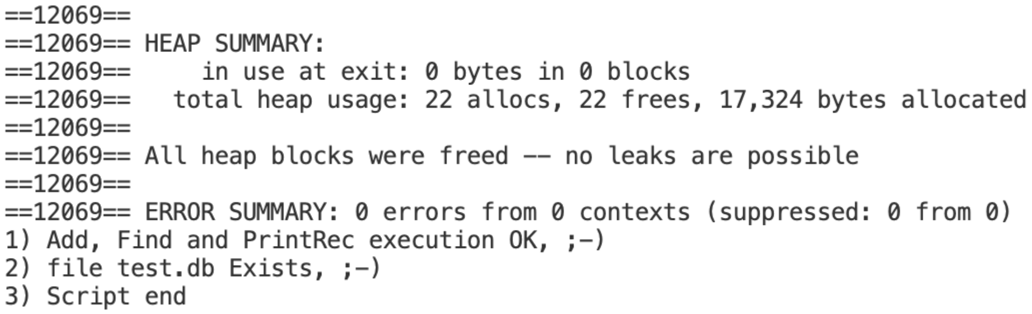
```
==11349==
==11349== HEAP SUMMARY:
==11349==     in use at exit: 0 bytes in 0 blocks
==11349==   total heap usage: 22 allocs, 22 frees, 13,176 bytes allocated
==11349==
==11349== All heap blocks were freed -- no leaks are possible
==11349==
==11349== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
1) Delete index records plus list of deleted books OK, ;-)
2) file test.ind Exists, ;-)
3) delete books files are identical, ;-)
4) Script end
```

- test_use_deleted_books.sh



```
==11693==
==11693== HEAP SUMMARY:
==11693==     in use at exit: 0 bytes in 0 blocks
==11693==   total heap usage: 22 allocs, 22 frees, 17,264 bytes allocated
==11693==
==11693== All heap blocks were freed -- no leaks are possible
==11693==
==11693== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

- test_find_printrec.sh



```
==12069==
==12069== HEAP SUMMARY:
==12069==     in use at exit: 0 bytes in 0 blocks
==12069==   total heap usage: 22 allocs, 22 frees, 17,324 bytes allocated
==12069==
==12069== All heap blocks were freed -- no leaks are possible
==12069==
==12069== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
1) Add, Find and PrintRec execution OK, ;-)
2) file test.db Exists, ;-)
3) Script end
```