CS3743 Program #2 Hash File Using Chaining for Synonyms (40 points)

© 2018 Larry W. Clark, this document may not be copied to any other website.

This is the **second part** of a two part programming assignment. In this part, we are implementing chaining to handle synonyms.

I have provided a driver program, include file, and input file. The driver is provided to reduce your effort on this programming assignment.

cs3743p2Driver.c - driver program which invokes your functions. It also provides a hash function.

cs3743p2.h - include file which contains constants, HashHeader and Book typedefs, and function prototypes.

p2Input.txt - stream input file used by the driver to specify what needs to be invoked

p2ExtraInput.txt – used for the extra credit input

Those files are provided at **/usr/local/courses/clark/cs3743/2018Sp/P2** which you can access from a fox server.

This program will use your code from the first programming assignment, making changes to some of the functions, and adding some new ones.

Functions you must code:

int **hashCreate**(char szFileNm[], HashHeader \*pHashHeader) – reused from program #1

FILE \***hashOpen**(char szFileNm[], HashHeader \*pHashHeader) - reused from program #1

int **readRec**(FILE \*pFile, int iRBN, void \*pRecord, int iRecSize) - reused from program #1

int **writeRec**(FILE \*pFile, int iRBN, void \*pRecord, int iRecSize) – reused from program #1

int **bookInsert**(FILE \*pFile, HashHeader \*pHashHeader, Book \*pBook) – you must modify this

This function inserts a book into the specified file.

* Determine the RBN using the driver's hash function.
* Use readRec to read the record at that RBN.
* If that location doesn't exist or the record at that location has a szBookId[0] == '\0':
  + Sets its iNextChain to 0.
  + Write this new book record at that location using writeRec.
* If that record exists and that book's szBookId matches, return RC\_REC\_EXISTS. (Do not update it.)
* Otherwise, it is a synonym to the book in the primary area:
  + Determine if it exists in the synonym chain. If it does already exist, return RC\_REC\_EXISTS. (Do not update it.)
  + Otherwise, insert it in the overflow area and set the **last record** in its chain to point to this new one. Note that if this is the first synonym, the primary record's iNextChain should point to this new record.
  + When inserting onto the end of the synonym chain, make certain you re-write that last chain record, changing its iNextChain.

int **bookRead**(FILE \*pFile, HashHeader \*pHashHeader, Book \*pBook) – you must modify this

This function reads the specified book by its szBookId.

* Determine the RBN using the driver's hash function.
* Use readRec to read the record at that RBN.
* If the book at that location matches the specified szBookId, return the book via pBook and return RC\_OK.
* Otherwise, it is a synonym to the book in the primary area:
  + Determine if it exists in the synonym chain. If it does exist, return the book via pBook and return RC\_OK.
  + Otherwise, return RC\_REC\_NOT\_FOUND.

int **bookUpdate**(FILE \*pFile, HashHeader \*pHashHeader, Book \*pBook)

This function reads the specified book by its szBookId. If found, it updates the contents of the book without losing the record's iNextChain. If not found, it returns RC\_REC\_NOT\_FOUND.

int **bookDelete**(FILE \*pFile, HashHeader \*pHashHeader, Book \*pBook)

If you did not do the extra credit, create a simple function that just returns RC\_NOT\_IMPLEMENTED.

**Note**:

1. Your code must be written based on my programming standards and placed in cs3743p2.c
2. Do not modify either cs3743p2.h or cs3743p2Driver.c.
3. You must run your code on a **fox** server.
4. Turn in your **cs3743p2.c** and **p2out.txt** (output) using a zip file named (*LastFirst*.zip) via BlackBoard.
5. In the notes in BlackBoard, also specify whether you did the extra credit program.

**Extra Credit** (10 pts + 200 / N)

* To be **eligible** for extra credit your code must meet **all** the requirements including the programming standards and your solution **must not be late**.
* N is the number of students who met all the requirements which is a combination from both sections of students.
* Implement the ability to delete books from the hash file. When a book is deleted, we simply zero out all of its information except (of course) iNextChain.

int **bookDelete**(FILE \*pFile, HashHeader \*pHashHeader, Book \*pBook)

* A deleted book will **not be removed from the synonym chain**. It will leave an item in the primary area or on the synonym chain with an empty szBookId.
* This will impact your code for **bookRead**:
  + You may encounter empty szBookId values along a synonym chain (including in the primary area). This doesn't mean that the book wasn't found. Instead, it might be later in the chain.
* This will impact your code for **bookInsert**:
  + Similar to bookRead, you must look throughout a synonym chain regardless of whether an empty szBookId is encountered.
  + If the book doesn't already exist in the synonym chain, place it in the **first empty slot** along the chain if there is one. (This might be in the primary area or in the overflow area.) If there isn't an empty slot, it will have to be inserted as a new overflow synonym at **the end** of its chain.
* Run your code using **p2ExtraInput.txt**.

**Sample Output (partial):**

\* CS3743 p1Input.txt

\* Nuke the Hash file if it exists

>> NUKE BOOK book.dat

\* Opening of a non-existent file should cause an error

>> OPEN BOOK book.data

\*\*\*\* ERROR: File does not exist or has invalid format: 'book.data'

\* Create the hash file

>> CREATE BOOK book.dat 19

Record size is 84

>> DUMP BOOK book.dat

Primary=19, HighOverflow=19, RecSize=84

\* Creating it again should cause an existence error

>> CREATE BOOK book.dat 17

Record size is 84

\*\*\*\* ERROR: file already exists

\* Open it

>> OPEN BOOK book.dat

\* #1 Insert some books

>> INSERT BOOK JOYPGM001,The Joys of Programming,TECH,PGMING,100

Hash RBN is 8

>> DUMP BOOK book.dat

Primary=19, HighOverflow=19, RecSize=84

8 Next=0

JOYPGM001 TECH PGMING 100 The Joys of Programming

>> INSERT BOOK PYTHON001,Learn Python Without Getting Bit,S PRESS,PGMING,200

Hash RBN is 1

>> DUMP BOOK book.dat

Primary=19, HighOverflow=19, RecSize=84

1 Next=0

PYTHON001 S PRESS PGMING 200 Learn Python Without Getting Bit

8 Next=0

JOYPGM001 TECH PGMING 100 The Joys of Programming

>> INSERT BOOK SQLDBB001,Making Your DB Queries SQueeL,XYZ,DB,300

Hash RBN is 16

>> INSERT BOOK TECHDR001,My Technical Dream Job,TECH,PGMING,400

Hash RBN is 18

>> INSERT BOOK PERLLL001,Is PERL the Jewel of Programming,S PRESS,PGMING,500

Hash RBN is 16

>> DUMP BOOK book.dat

Primary=19, HighOverflow=20, RecSize=84

1 Next=0

PYTHON001 S PRESS PGMING 200 Learn Python Without Getting Bit

8 Next=0

JOYPGM001 TECH PGMING 100 The Joys of Programming

16 Next=20

SQLDBB001 XYZ DB 300 Making Your DB Queries SQueeL

18 Next=0

TECHDR001 TECH PGMING 400 My Technical Dream Job

Overflow:

20 Next=0

PERLLL001 S PRESS PGMING 500 Is PERL the Jewel of Programming

\* #2 Read an existing book

>> READ BOOK TECHDR001

.. Next=0

TECHDR001 TECH PGMING 400 My Technical Dream Job

\* #3 Read should not find this one

>> READ BOOK JAVADD001

\*\*\*\* ERROR: record not found

\* #4 Insert an existing book - should cause an error

>> INSERT BOOK TECHDR001,My Technical Dream Job Again,TECH,PGMING,444

Hash RBN is 18

\*\*\*\* ERROR: record already exists

>> READ BOOK TECHDR001

.. Next=0

TECHDR001 TECH PGMING 400 My Technical Dream Job

\* #5 Insert some books and some synonyms

>> INSERT BOOK JAVADD001,Java Isn't an Addiction,S PRESS,PGMING,600

Hash RBN is 2

>> INSERT BOOK LINUXX004,Learning Linux,XYZ,OS,700

Hash RBN is 10

>> INSERT BOOK COBOLL001,How Your Grandpa Coded in COBOL,S PRESS,PGMING,800

Hash RBN is 19

>> INSERT BOOK EXCELL001,Excel at Excell,MS PRESS,ACCOUNT,200

Hash RBN is 2

>> INSERT BOOK PRANKS001,Software Pranks,TECH,PGMING,300

Hash RBN is 1

>> DUMP BOOK book.dat

Primary=19, HighOverflow=22, RecSize=84

1 Next=22

PYTHON001 S PRESS PGMING 200 Learn Python Without Getting Bit

2 Next=21

JAVADD001 S PRESS PGMING 600 Java Isn't an Addiction

8 Next=0

JOYPGM001 TECH PGMING 100 The Joys of Programming

10 Next=0

LINUXX004 XYZ OS 700 Learning Linux

16 Next=20

SQLDBB001 XYZ DB 300 Making Your DB Queries SQueeL

18 Next=0

TECHDR001 TECH PGMING 400 My Technical Dream Job

19 Next=0

COBOLL001 S PRESS PGMING 800 How Your Grandpa Coded in COBOL

Overflow:

20 Next=0

PERLLL001 S PRESS PGMING 500 Is PERL the Jewel of Programming

21 Next=0

EXCELL001 MS PRESS ACCOUNT 200 Excel at Excell

22 Next=0

PRANKS001 TECH PGMING 300 Software Pranks

\* #6 READ one that is on chain

>> READ BOOK PRANKS001

.. Next=0

PRANKS001 TECH PGMING 300 Software Pranks

\* #7 Insert some more

>> INSERT BOOK ARTINT001,A.I. Practical Algorithms,S PRESS,PGMING,400

Hash RBN is 4

>> INSERT BOOK PYTADW001,Programming in Python,ADWES,PGMING,500

Hash RBN is 11

>> INSERT BOOK FUNDBS001,Fundamentals of Database Systems,PEARSON,DB,900

Hash RBN is 7

>> INSERT BOOK COMPILER1,Compilers,PEARSON,PGMING,1100

Hash RBN is 7

>> DUMP BOOK book.dat

Primary=19, HighOverflow=23, RecSize=84

1 Next=22

PYTHON001 S PRESS PGMING 200 Learn Python Without Getting Bit

2 Next=21

JAVADD001 S PRESS PGMING 600 Java Isn't an Addiction

4 Next=0

ARTINT001 S PRESS PGMING 400 A.I. Practical Algorithms

7 Next=23

FUNDBS001 PEARSON DB 900 Fundamentals of Database Systems

8 Next=0

JOYPGM001 TECH PGMING 100 The Joys of Programming

10 Next=0

LINUXX004 XYZ OS 700 Learning Linux

11 Next=0

PYTADW001 ADWES PGMING 500 Programming in Python

16 Next=20

SQLDBB001 XYZ DB 300 Making Your DB Queries SQueeL

18 Next=0

TECHDR001 TECH PGMING 400 My Technical Dream Job

19 Next=0

COBOLL001 S PRESS PGMING 800 How Your Grandpa Coded in COBOL

Overflow:

20 Next=0

PERLLL001 S PRESS PGMING 500 Is PERL the Jewel of Programming

21 Next=0

EXCELL001 MS PRESS ACCOUNT 200 Excel at Excell

22 Next=0

PRANKS001 TECH PGMING 300 Software Pranks

23 Next=0

COMPILER1 PEARSON PGMING 1100 Compilers