Hayden Thrasher and Sandy Becerra

Database

Susan Gauch

9/11/16

Assignment 1: Report

For this assignment students were asked to create a program that would read a file named input.txt and perform several tasks with it. Students were allowed to partner up for this assignment.The structure of this assignment was designed so as to resemble a system that would allow someone to open an existing database and be able to perform the following tasks: modify an entry, delete an entry, add an entry, or close the database. Users should also be able to create a new database and perform the stated tasks for it as well.

To begin, we started by making the "skeleton" of the code. We first made sure that the program would be able to provide the user with a good and readable menu. Our program consisted of 2 menus. The first allows the user to chose from the following options: open a database, create a new database, or exit the program. If the user choose to open a database then they were taken to a second menu were a list of possible tasks to perform would be chosen from(this is the list mentioned in the first paragraph). After having the "skeleton" done we continued by creating functions for the listed tasks and implementing them as we progressed.

Adding an Entry: This task was among one of the easier ones to implement. Given that students had been provided with the binarySearch() function, adding an entry simply consisted of asking the user for input. We used binarySearch to make sure that users were not adding an entry with the same id number as one that was already in the file.

Initially the user is asked how many entries they wish to add and according their response a for loop will begin and ask for input from the user x amount of times (x being the number of entries the user chose to enter). Our program first writes the new entries to a temp file and then appends them to the original database and sorts it(this was done by using unix commands cat and sort). To keep track of the correct number of records within a file, we had a variable named DATOS that would be modified in our program where necessary and when the user closed the database, DATOS was written to a file called "numero.txt". "numero.txt" gets read in when the user opens the database. The tricky part with this function was getting the format to match the original file.

Deleting and Modifying an entry: In order to delete or modify data, we had to conjure some way to get the cursor in the right position so as to modify or delete the right entry. In order to do this we first had to read in the lengths of each field according to input.txt. Once these fields(ExperienceColumnLength, IDColumnLength, etc) were read in then we proceeded by asking the user which field they wished to modify/delete. If delete, then cursor was moved to right position according to field length and default values were placed in (if modify then user input was placed in). To modify we created a switch statement.

Search: students were provided with the binarySearch function

Creating a new database: In order to allow the user to create a new database, we had to write to a completely new file, similar to writing to a temp file. The most difficult part about implementing this function was formatting. Even though were able to write to the file without trouble, we were required to make sure that the format in the new database was the same as that of input.txt. If formatting was off, then binarySearch failed and completing other tasks with this new database became impossible. Similar to that of deleting or modifying an entry, we had to first move the cursor to the right position in the file and then write to the file. This lead for proper formatting to be accomplished and made sure that it matched that of input.txt

***Please look at Test_Cases file in the zip to read about test conducted ***

Because of our experience programming, reading and writing to files was not that difficult. Some of the more difficult and time consuming implementations were deleting/modify the database. Because of having to move the cursor to the right position this lead to several attempts being made and making sure it functioned well with other parts of the program. Getting format right was the most time consuming of all. Even though format looked correct to us and compared right to input.txt, we still had some issues with binarySearch. At first this was a problem when users wanted to search for added entries, but after some time we managed to get format so that it was correctly read in. We were not so successful when trying to search the newly created database. Even though the format looks right by the eye, the program doesn't seem to read it in right and therefore was unable to complete a correct search.