

CS4350

Unix Sys Programming - Project 1

Dylan Wagner

Tiffany Connors

Arthur Reeves

Matthew Learned

Section I (Introduction):

This project was done remotely without a group meetup face to face. The individual's schedules made it tough to meet up throughout the project, so the coding was done using a Github repo to collaborate on the code and keep everyone up to date with the newest commits and changes. Dylan coordinated the bulk of the project. Although not meeting face-to-face, Dylan communicated with the team and pushed certain deadlines while helping to keep the team organized and working as a group.

Dylan wrote the User Menu, setup the database text file and created a skeleton of the project putting the functions in place without their implementation. He also wrote the add function and assisted Tiffany in testing, debugging and QA. Tiffany wrote a majority of the code and provided most of the conceptual ideas for function implementation like some findRecord ideas as well as ideas for the menu. Tiffany implemented the removeRecord, displayData, and updateRecord functions. These functions remove a specific record, display the database file (a useful function for debugging), and update a specific record, respectively. Tiffany also played a large role in testing, debugging, and QA. Matthew added input validation and wrote original version of the findRecord function. Arthur added to the partial implementation of the findRecord to meet the acceptance criteria and give more performance.

Dylan wrote the introduction for the report as well as Section II, describing the database and text file. Tiffany wrote Section III of the report, describing each function and showing an update record operation step-by-step. Arthur wrote section IV of the report, describing each line of the function. Matthew wrote the beginning portion of section III, describing the overall functionality and interface, as well as describing the user menu.

Section II (Task I):

The design of the database and text file is modeled after the example given by Dr. Gu. We are using the character ":" as the IFS. The database is organized by Name, Address, Phone, and then Email. These fields hold a name of a person, an address of residence, a phone number, and an email address, respectively. A sample database with four records follows:

Name:Address:Phone:Email

John:1234 Doe Ln:5123140987:john.doe@example.com

Jane:2345 Miroad Rd:5121234567:jane.doe@gmail.com

Jim Smith:543 Country Ln:2819350978:jim_smith@yahoo.com

Kara Wheeler:935 Deats Way:3619057896:kara_wheeler123@gmail.com

Section III (Task II):

Program Interface

The program is presented through the terminal, and interacted with through bash. Users interface with the program through the stdin, and all information is printed to stdout.

Main Menu

Once the program is started, users are greeted with a welcome message, and prompted to select one of the following options:

- a) Find a record
- b) Add a new record
- c) Update a record
- d) Remove a record
- e) Display a record
- f) Quit

The user selects their option by pressing the corresponding key and enter. If an invalid option is selected, then the user is once again prompted to enter a valid option. Once a valid option is selected, an appropriate function is called to handle their request, and once that function is finished users are returned to the main menu.

findRecord()

This function searches the database file for a pattern matching the user input. If a pattern is found on any line, it will be displayed to the screen in a formatted table.

addRecord()

This function adds an entry to the database by getting input from the user for each field (name, address, phone, email). The function adds the appropriate file separator, ':', between the fields and then inserts the new record at the end of the database file.

updateRecord()

This function enables the user to update the contents of a record in the database, one field at time, for a single record. If no records or multiple records matching the user's search term are found, the function displays an error message and returns to the main menu.

removeRecord()

This function enables the user to remove a record from the database. If no records or multiple records matching the user's search term are found, the function displays an error message and returns to the main menu.

displayData()

This function displays the contents of the database file to the screen. This function also calls all previously stated functions through a case statement which the user provides the input for. Also providing an option to exit the program and error checking if the user provides a invalid input.

Updating a record:

Step 1. The user is prompted to enter the record he/she wants to update.

```
Enter the record you would like to update:
```

Step 2. The program searches the database for a record that contains text that matches the user's input.

Step 3.a. If multiple records are found that match the user's input, the found records and an error message are displayed. The user is returned to the main menu.

```
Enter the record you would like to update:
Gu
Qijun Gu:209E Comal:2453518:qijun@txstate.edu
Dr Gu:209 Comal:2453518:qijun@txstate.edu

Error: Found multiple records. Cannot update.
Try again with a more specific search.
```

Step 3.b. If a single record matching the user's search query is found, the record is displayed.

```
Enter the record you would like to update:
Bob
6:Bob:123 A Street:7654321:bob@yahoo.com
```

Step 4. A menu is displayed to the user, prompting him/her to select which field to update.

```
Which field would you like to update:
1. Name
2. Address
3. Phone
4. Email
5. Done
```

Step 5. The user is prompted to enter the new information.

```
Enter new name:
Bobby

Which field would you like to update:
1. Name
```

Step 6. The update menu is displayed in a loop until the user selects that he/she is done updating the record.

```
Which field would you like to update:
1. Name
2. Address
3. Phone
4. Email
5. Done
5

--- Record has been updated! ---

Select from the options below...
(a) Find a record
(b) Add a new record
(c) Update a record
(d) Remove a record
(e) Display the database
(f) Quit
> Selection is :

```

Step 7. The program removes the original record and appends the updated record as the last entry in the database.

Step 8. A success message is displayed and the user is returned to the main menu.

Section IV (Task III):

```
findRecord() {
    echo "Enter the data to search for:"
    #Prompts the user for input to search
    read searchData
    #Reads the input into searchData
    printf "%-20s%-20s%-20s%-20s\n" "Name" "Address" "Phone"
    "Email"
    #Prints a format labeling the information
    while IFS= read -r line || [[ -n "$line" ]]; do
    #While reading do(makes sure last line of file is read)
        if echo $line | grep -iq "$searchData"; then
    #If searchData is found in a round then
            oIFS=$IFS
    #Saves a copy of IFS
            IFS=":"
    #Sets the IFS to ':'
            set -- $line
            for w in $@
    #for every field in the record do
            do
                printf "%-20s" $w
    #prints the record data in the same format
            done
            printf "\n"
            IFS=$oIFS
    #Sets IFS back to original
        fi
    #end of condition statement
    done < database.txt
    #Redirects content in database.txt to stdin for loop

    echo ''
    #Prints ''
}
}
```

1) Find an exactly matching record for a query

```
Enter the data to search for:
gu
Name          Address          Phone          Email
Qijun Gu      209E Comal        2453518        qijun@txstate.edu

Select from the options below...
(a) Find a record
(b) Add a new record
(c) Update a record
(d) Remove a record
(e) Display the database
(f) Quit
> Selection is :
```

2) Find multiple matching records for a query

```
Enter the data to search for:
d
Name          Address          Phone          Email
Name          Address          Phone          Email
Qijun Gu      209E Comal        2453518        qijun@txstate.edu
Apan Qasem    Comal 307A        1111111        aq10@txstate.edu
Eric          889 RoadName      9999999        eric@hotmail.com
NewName       NewAddress        1231234        newEmail@gmail.com
```

3) Find nothing for a query

```
Enter the data to search for:
122222
Name          Address          Phone          Email

Select from the options below...
(a) Find a record
(b) Add a new record
(c) Update a record
(d) Remove a record
(e) Display the database
(f) Quit
> Selection is :
```