UCP Assignment

by Kei Sum Wang
student ID: 19126089

Contents

The C file components of the program

- tvguide.c – contains main function.
- interface.c – contains functions that handles terminal input and output.
- fileIO.c – contains function that deal with reading and writing to file.
- linkedList.c – contains function dealing with creation, insertion, manipulation, accessing, destroying a linked list.
- Comparison.c – contains functions that handle comparison of data, functions are implemented for qsort.

Corresponding header files

- interface.h
- fileIO.h
- linkedList.h
- comparison.h
- tvguide.h

Other header files

- boolean.h – contains TRUE and FALSE constants, constant EQUALS has also been added for better readability when using functions such as strcmp().
- Data.h – contains structs required for TV guide entry

By Kei Sum Wang
student ID: 19126089

Description and purpose of implemented functions

tvguide.c functions

main

main is the entry point of the program, it takes in 2 command arguments first being the name of the input file and second being the name of the output file. Main checks whether the file names have been entered, if the file names haven't been entered then program informs user to input an input file and output file, otherwise main calls a function called process() to begin processing of tv guide entries.

process

Imports the command argument vector from main, containing file names for input and output.
Purpose of process is to begin creation of the data structures and handle the flow events of reading, user input, sorting and writing and displaying tv guide entries by calling the necessary functions from other c files to handle these tasks. This is also implemented to make main look tidy.

sortGuides

Imports an array containing tv entries, string for sort type and length of array.
Purpose of the sortGuides is to choose a qsort implementation appropriate for the specified type of sort given by user, either time or name. After the selected qsort function is done the array should be sorted in the chronological order(time) or alphabetical order(name)

filterByDay

Imports array of sorted tv entries, string for day, length of sorted tv entries and an integer pointer to filterLength.
Purpose of this function is to filter entries of specified day in the imported array, if entries are not of specified day then the entry will be removed from array. The end result of the imported array will only contain entries of specified day.

By Kei Sum Wang
student ID: 19126089

Interface.c functions

dayInput

Imports string day
Purpose of dayInput is to get user to enter a day, the day will be used for filtering purpose for displaying and writing of entries on specified day. If the day is not valid the program will keep prompting until the user enters a valid day.

sortInput

Imports string sort
Purpose of sortInput is to get user to enter a sort, the sort type specified will be used for sorting purpose to indicate chronological sorting or alphabetical sorting of guides. If the sort input is not "time" or "name" the program will keep prompting until the user enters a valid sort.

CheckDayInput

Import string day
Purpose of checkDayInput is to check if the entered day is a valid day of the week using c strcmp function, if the day match a day of the week then checkDayInput should return 0 which indicates match or any other integer to indicate not match.

upperFirstLetter

Import a string
Purpose of upperFirstLetter is to convert First letter to uppercase and the rest of following letters to lower case.
Example – "mONday" converts to "Monday"
note: upperFirstLetter depends on convertLower for the lowercase conversions

convertLower

Import a string
Purpose of convertLower is to convert every character in string to lowercase.
Example - "TIME" converts to "time"

display

Import array of TV entries, length of array (Integer),
strings day and sort
Purpose is to display TV entries of selected day and sort from an array to the screen.

By Kei Sum Wang
student ID: 19126089

fileIO.c functions

storeFileContent

Imports string fileName and Linked List
Purpose of storeFileContent is to read in file and store TV entry details into structs which are then inserted into imported linked list.

writeToFile

Imports string fileName, array of TV entries
and length of array(Integer)
Purpose of writeToFile is write tv entries of imported array to specified file.

validateFormat

Imports FILE pointer
Purpose of validateFormat is to read the file and check if each line is in the correct format, validateFormat specifically checks if line contains a string and the next line contains a valid day and time, this is repeated in this order until there are no lines to read. If the file doesn't exist it is considered invalid. ValidateFormat returns TRUE(any Integer other than 0, defined by boolean.h) if all lines found in correct format otherwise FALSE(0, defined by boolean.h).

By Kei Sum Wang
student ID: 19126089

comparison.c functions

compareTime

Imports constant pointers(void) x and y
Purpose of compareTime is to compare two time fields from TV entry struct. Checks if one of the times are larger(function returns 1), equal(function returns 0) or smaller than the other - (functon returns -1).

compareName

Imports constant pointers(void) x and y
Purpose of compareName is to compare two name fields from TV entry struct. This function utilises c string function strcmp to compare the names which returns a positive integer if string is greater than other , 0 if equal or negative integer if less than the other. compareName returns strcmp's returns values.

By Kei Sum Wang
student ID: 19126089

linkedList.c function

createLinkedlist
Imports nothing
Purpose of createLinkedList is to create and initialise linked list. CreateLinkedList returns a created list.

insertFirst
Imports Linked List and a pointer(void) to data
Purpose of insertFirst is to insert data at the front of a list.
Note: This is also known as stacking.

insertLast
Imports Linked List and a pointer(void) to data
Purpose of insertLast is to insert data at the end of a list.
Note: This is also known as enqueueing.

insertMid
Imports Linked List and a pointer(void) to data
Purpose of insertMid is to insert data at the middle of a list.

deleteFirst
Imports Linked List
Purpose of deleteFirst is to delete data from the front of a list. DeleteFirst returns the value of data deleted from the front of list.

deleteLast
Imports Linked List
Purpose of deleteLast is to delete data from the end of a list. DeleteLast returns the value of data deleted from the end of list.

deleteMid
Imports Linked List
Purpose of deleteMid is to delete data from the middle of a list. DeleteLast returns the value of data deleted from the middle of list.

getLength
Imports Linked List
Purpose of getLength is to iteratively traverse the linked list from front to end counting the number of existing nodes. getLength returns the length of list.

By Kei Sum Wang
student ID: 19126089

getElement

Imports Linked List and an index(Integer)
Purpose of getElement is to iteratively traverse and count linked list nodes until number of counts reach position specified by index. Data(void pointer) at index position is returned.

peekFirst

Imports Linked List
Purpose of peekFirst is to see what is at the front of the list.
PeekFirst return data(void pointer) of node at front of list.

peekMid

Imports Linked List
Purpose of peekMid is to see what is at the middle of the list.
PeekMid return data(void pointer) of node at middle of list.

peekLast

Imports Linked List
Purpose of peekLast is to see what is at the end of the list.
PeekLast return data(void pointer) of node at end of list.

displayList

Imports Linked List and pointer to a function
Purpose of displayList is to iteratively traverse and print out contents of each node until there are no nodes.
Note: the pointer to a function is intended to point to a function that implements an approriate way to print the data.

storeIntoArray

Imports a Linked List, array of pointers(void pointer to pointer) and length of array(Integer)
Purpose of storeIntoArray is to copy contents of Linked List into imported array.

freeLinkedList

Imports Linked List and dataFree(Integer)
Purpose of freeLinkedList is destroy and free all nodes of the linked list.
If necessary the nodes data can be freed if the import value of dataFree is any integer other than 0.
FREE_DATA (value 1) and DONT_FREE_DATA (value 0) are macros defined in the linkedList.h

By Kei Sum Wang
student ID: 19126089

The approach of filtering TV guide entries

Once the file has been read into a linked list, user has selected a day and a sort type. All TV entries of linked list will be copied to an array, then the entries in the array are sorted in chronological order or alphabetical order. After sorting the array, the next step is to filter the array based on specified day.

Filtering was implemented by using a for loop(for loop used integer variable j to index and i is a separate counter used in an if statement, both start from 0), looping through all entries until the end of the sorted array. During the loop, an if statement is implemented to check if the entries day matches the day selected by the user, if the selected day and entry day matches then the entry of current jth position is shifted to ith position then the i counter is incremented. If an entry's day does not match day specified then that entry will be removed from the guide. If statement repeats until end of array. This overall results in a guide array with entries only of specified day.

This implementation can be found in the tvguide.c file, the function is called filterByDay which filters the guide to only contain entries of specified day.
Page 2 of this report has more information about filterByDay.

The Alternative approach
Instead of storing all the TV Entries that were read into the linked list into an array, sorting the array then filtering out the array. Another approach could be to traverse the linked list and retrieve the entries in the list, checking if the entries contain the same day as selected then copy the entries to an array that contains filtered entries. After all the entries of selected day are stored into the array from the linked list, the selected sort (time or name) can be performed on the filtered array.

This alternative approach may have been better than the current implementation since there is no need to create and sort an array that stores all the entries from the entire linked list. Filtering out entries from linked list into an array then sorting would have been more efficient since we directly retrieve specific entries and there is no need remove entries irrelevant of day from the guide.

By Kei Sum Wang
student ID: 19126089

Demonstrate program with expected input and output files

Input file format

The input file should contain TV shows starting at first line of the file, the format of the TV shows details should start with the name of the show followed by the day and time on a newline just after the name.
The file cannot contain an empty new line between TV shows and its details, program and validation expects the day and time on the line after the name of the show. The line after the day and time may contain another TV show repeating the format as above, this line also cannot be a empty new line.

Example of Input file with 4 entries.

```
The Walking Dead
Saturday 19:30
Rick and Morty
Saturday 07:30
Silicon Valley
Tuesday 19:30
The Matrix
Tuesday 10:15
```

Running the program

The command-line to exectute the program is-
./program [inputfilename] [outputfilename]

note: A file to read and a file to write to must be provided in command-line to begin processing of tv entries, otherwise program will tell user that they need to provide the file names in the command line.

User input via terminal

Once the command-line has successfully been executed and file has been read into a linked list, the user will prompted enter a day followed by a sort. The day must be from Sunday to Saturday and the sort must be either time or name.
The input is not case sensitive given that the day and sort are valid.
Example1 “SaTUrdAy” is still valid since it is a valid day of the week and it will be converted to “Saturday”.

By Kei Sum Wang
student ID: 19126089

Example2 Sort “nAMe” will converted to “name”

Outputting to screen

Once the TV entries have been sorted alphabetically or chronologically and filtered based on selected day into an array, entries in the array(filtered TV guide) will be outputted to screen.

Example1 screen output with Saturday and time sort selection.

```
==== Shows on Saturday ====
==== Sort by time =========
07:30 - Rick and Morty
19:30 - The Walking Dead
```

Example2 screen output with Tuesday and name sort selection

```
==== Shows on Tuesday ====
==== Sort by name =========
19:30 - Silicon Valley
10:15 - The Matrix
```

Writing to file

filtered TV guide will be written to a file specified by user.

Example1 contents written to file with Saturday and time sort selected

```
07:30 - Rick and Morty
19:30 - The Walking Dead
```

Example2 of contents written to file with Tuesday and name sort selected

```
19:30 - Silicon Valley
10:15 - The Matrix
```


By Kei Sum Wang
student ID: 19126089

Example of the program running with valgrind sucessfully

```
==26451== Memcheck, a memory error detector
==26451== Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward
et al.
==26451== Using Valgrind-3.8.1 and LibVEX; rerun with -h for
copyright info
==26451== Command: ./program reporteg.txt out.txt
==26451==
Enter a day of the week:
Tuesday
Sort by time or name:
name

==== Shows on Tuesday ====
==== Sort by name =========
09:30 - Silicon Valley
21:15 - The Matrix

Write to file: out.txt success
==26451==
==26451== HEAP SUMMARY:
==26451==     in use at exit: 0 bytes in 0 blocks
==26451==   total heap usage: 13 allocs, 13 frees, 1,800 bytes
allocated
==26451==
==26451== All heap blocks were freed -- no leaks are possible
==26451==
==26451== For counts of detected and suppressed errors, rerun
with: -v
==26451== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 8
from 6)
```


By Kei Sum Wang
student ID: 19126089