CSCI 423: Writing Report

Due on Thursday, Dec. 1, 2016

Wang 11:00am TTh

Xavier Pittman

Contents

Report 2

Part A 2

Part B 2

Software Design 2

Results Expected 3

Report

Part A

The **Project_PartA.c** file is the source code for the "Part A" portion of the Project. It must first be compiled with the pthread library linked. After the code is compiled, it requires a text file name as the second input argument simultaneously with execution. If there are not the correct amount of arguments at execution the program will generate an error. If the text file name that is provided doesn't exist the program will terminate. Once executed, the program will open the text file and find the frequency of words used in the text file. It will then sort the words alphabetically.

Part B

The **Project_PartB.c** file is the source code for the "Part B" portion of the Project. It must first be compiled with the pthread library linked. After the code is compiled, it requires a text file name as the second input argument and an integer as the third input argument simultaneously with execution. This integer is used to instruct how many threads should be used. If there is not an argument passed, too many arguments passed, or an integer less than 1 passed when program is executed and error message will display and the program will terminate. If an acceptable argument is passed to main, the program will open the text file and find the frequency of words used in the text file. It will then sort the words alphabetically.

Software Design

Libraries

There are six libraries that are included in the source code. These libraries include **studio.h, pthread.h, stdlib.h, string.h, ctype.h, & math.h**. The **stdio.h** library is necessary because it provides gives the ability to use different variable types. It also allows different input and output functions. The **pthread.h** library makes it easy to implement multiple threads in the program. It provides many functions that simple to create threads, find attributes of threads, and join threads. The previously mentioned functions were vital in this multi-threaded project. The **stdlib.h** library, similar to the stdio.h, provides different variable types, macros, and key functions as well. This library make it simple allocating memory for certain variables by providing functions that perform this task. It provides necessary functions that allows the programmer to operate with files. The **string.h** library has functions necessary to simplify procedures that involve strings. It provided functions that allowed for strings to be compared to each other. This was necessary because it allowed for an easy way to sort words from the file in alphabetical order by simply using a function.

Report [Software Design] continued on next page...

The **ctype.h** library provided features that make it possible to distinguish whether a letter is uppercase or lowercase.

Functions

This program has seven functions. Each of these functions provide a key ingredient that contributes towards the program successfully performing as expected. These functions include **swap, bubbleSort, getWord, addWord, Display, printList**. The **swap()** functions takes in two structs. It takes the two parameters from the "word" struct and swaps them. The**bubbleSort()** functions utilizes the **swap()** function. The **bubbleSort()** requires a "word" struct. It compares the "str" variables of the current struct in the linked list and the next struct in the linked list to see where they rank alphabetically. If they are positioned correctly according to rank then nothing happens. If they are not positions correctly, they are sent to **swap()** where the contents of the two structs are swapped. The **getWord()** function requires a text file, char, and a buffer size. This function was designed primarily to get words from the text file. The **addWord()** function adds the word to a "word" struct after it has been retrieved from the file. It check whether the word is new or whether it has been added already. If the word has been added to a struct already, the function will increment the structs frequency. Regardless of whether the was has already been used or not, the total amount of words is incremented every time this function successfully adds a new word. The **printList() and Display()** funtions are essentially the same. The Display function traverses a linked list one at at time. The **printList()** function will traverse a linked list in its entire.

Global Variables There are three notable global variables and one typedef. These variables include **int wordCount, int differentwords, and char buf**. The typedef is **struct word**. The global variable **wordCount** is used to count total amount of words in the file. The global variable **differentwords** is used to count total amount of different words in the file. The global variable **buf** is used as a buffer. The typedef **struct word** is used as the link list structure. It contains two variables. The first variable is char str. This is used to store words from the file. The second variable is int freq. This is what is used to keep up with the frequency of a word. pNext is used to point to the next link in the linked list

Results Expected

After the programs are compiled and executed properly, the programs will utilize the functions above to count the words of a text file and display them in alphabetical order along with the words frequency.