Nathaniel Brown
CSC 412 - Assignment 2 - FS & IPC
9/24/2023 - (I apologize for the length of this - it's hard to keep it condensed without skipping)

Purpose:

The overarching purpose of this assignment is to develop individual c programs that accomplish specific data manipulation tasks via the usage of a custom read/write/open library. That data will be manipulated, and passed via output and piping methods to effectively reproduce the UNIX demo that was covered in lectures. The combined combination, and ordered execution demonstrate the power and usefulness of effective bash scripting. Below is the purpose of each program individually.

Fio Library:

- Write method in which the user will be able to write to an existing file.
- Create in which the user will be able to initialize a new empty file.

C Files:

- Sentence will read the unix_sentence text file and report it back to standard output.
- Makewords will effectively break each word into the sentence apart leaving one word per line.
- Lowercase will take all characters within a given input and return them as lowercase
- Sort will read from a file and return the words in an alphabetic order ascending from A-Z.
- Unique will cross compare strings within input and report back only the words that were given once.
- Mismatch will cross compare given words from standard input to those within a dictionary text file. The words that were not found within the dictionary will be printed to standard output.

Bash Scripts:

- For the sake of repetition; the bash scripts will effectively aim to achieve all of the above tasks within bash.
- Build will attempt to compile all files within the given directory, and provide them with the appropriate relevant custom libraries that we developed within another directory location.
- The demo file will compile all of the C files and pipe them together, ultimately dumping the combined output into a text file for grading. The result will be the combination of all of the C files in unison to produce the filtered and organized list of 'strange' terms/typos within the source text file we initially use in the sentence c/sh files.

Outline:

Fio Library:

- Create Function Method - Passed a char pointer that will be used as a file name; the return will be a new object of the file created.
- Write Function Method - will take two arguments: a pointer to the file type object and a pointer to the char data the user wishes to write to the file. The program will then write to the given file. Void function; no return.

C Files:

- Sort - a simple loop will handle the output of the given lines within main.
    - Bubble Sort Helper Function - will be passed a pointer to an array and an int representing the size of the array. Using the bubble sort methodology it will reorganize the passed array. No return will happen as all action is conducted within the function.
- Makelines - similar to before the the main output of this function will be handled within a loop in main (a while). Within that loop the following function will be built to effectively create new lines in between each word.
    - Makewords Helper Function - will be passed a char pointer and return a char pointer. The function will evaluate if the current character is a white space; if it is it will replace the character with a new line signifier. If not it will end and continue to be called within main until conclusion.
- Mismatch - should not need additional helpers. Within the primary loop orchestrating the evaluation of standard input, I will tokenize each word of input separated by spaces. From here I will have a nested while loop compare the token to the words within the dictionary. If it is found; the word we will break from the loop. If the word is not found; it is mismatched and will be reported to standard output.
- Unique - this function will handle looping through the data within a while loop and utilize a helper function to check if the current line is a duplicate or not.
    - Duplicate Line Checker Helper Function - will be passed a pointer to a given line, a pointer to an array of lines and an int tracking the number of lines in total. Looping through each line the function will compare the current line to the current line of the array. If true; the function will return that this line is a duplicate. Otherwise it will return it is false.

Bash Scripts:

- The main six bash scripts function in the same manner as the c files. (For the sake of redundancy and an ever growing document)
- Build Script - Will use the current working directory to check if there are C files within the directory. Using a conditional if they don't exist it will error and exit. Otherwise it will loop through each c files and compile them using the first part of their names as the executable name.
- Demo Script - This is mostly provided; but will first call mismatch and write the output into a new dictionary. Followed by executing and piping each of the c files into each other and outputting the data within an appropriate text file. The same will happen with the bash scripts.

Research:

For this assignment I believe the lab and lectures have left me pretty confident that after designing the layout for implementation within this document I am good to start implementing everything discussed so far. I can not think of anything in particular that I will need to do background research on. Most research will revolve around syntax with a new language (Bash / C). Albeit C is very similar to C++. I'm interested in what will come up within the Post Review for researched topics.

Fio Library and C Files:

- These will be pretty straightforward and should not require much background research.

Bash Scripts:

- Heavy syntax research will be required for this; the logic likely remains the same as the C implementations. I do not foresee issues on these.