Multi-process File Searcher Design Document

Author: Jacob Walton

Date: February 16, 2018

**Purpose:**

This is a program that is used to aid users in searching multiple files for a search word. This program was written to complete the project assigned in CIS 452 by Professor Wolfe.

**Overview:**

Users will enter the number of files to be searched followed by the filenames of the documents to be searched. The files must be in the same folder as the executable to be accessed. The user will then enter the word to be searched. The program will then search all of the files given by the user and report the word count of this word in each file. The program allows for the user to repeat the search process on as many words as desired. When the user is finished, the program can be exited using CRTL + C.

**Description of Flow/Data Structures:**

The program is initiated, and the Master process asks the user to enter the number of files (global integer) to be searched. The Master process then dynamically creates an array of character pointers to store n filenames. The Master asks the user to enter each file name which is stored in filenames array. Based on the number of files, 2 arrays are created to store n pipe file descriptors. 2 pipes per file name are created and stored in the 2 arrays created in the last step. The stats array which will be used by Master to store the word count info for each file is also created.

Then, for each filename, a process is spawned and sent to the Searcher function where the child process will live and die. Master then enters a loop where it will ask the user for a word to search for, send this word via each search pipe to each Searcher. The Master will read the word count from each stats pipe for each Searcher and report the word counts. Master repeats until a CTRL + C is entered.

Each Searcher opens the file name it is assigned by accessing the filenames array. Then the Searcher enters a loop where it will wait for the word to search for from the pipe connected to Master. Once a word is received, it will search its file, and report word count through the stats pipe back to Master. The Searcher then repeats the loop, waiting for a new word to search for until CTRL + C is entered.

When a CTRL + C is entered, a signal handler function is executed by each process (Master and Searchers) where all pipes are closed, and all memory is freed.

**Limitations/Assumptions:**

The program assumes the filenames entered are in the same directory as the executable. The program also assumes correct filenames are entered; it will exit automatically when an invalid filename is entered upon trying to open the invalid file. The logic for finding matches of a word is very limited. It will only match the exact casing of the word entered, and the word must be preceded and followed by a space.