Project Report
Name: Hannah Reinbolt
Date: 2/9/2021
Assignment: Project 2 – Pipe-based Wordcount Tool

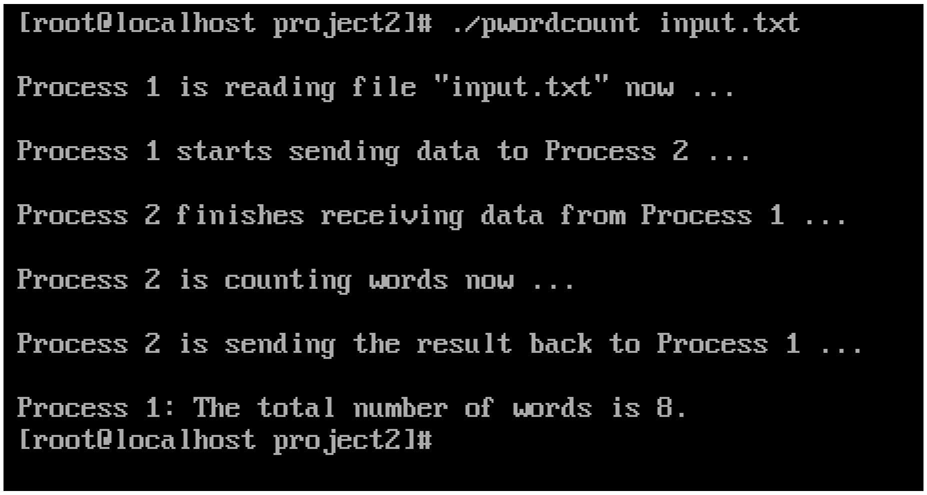
Design

The design for the pipe based wordcount tool consisted of three main parts; a helper file to store helper functions, a make file to compile quickly, and a main pwordcount file to build the unix pipes and utilize the helper functions to count the file words.
The main pwordcount file brought all the moving pieces together and the main purpose was to utilize unix pipes to count words from a user supplied word file. The helper file is meant to carry assistant functions for the pwordcount file. These functions include error messaging and the counting of words. The make file provided a quick way to compile both helper and pwordcount files together into a single executable using the gcc compiler. All development was done in the linux CentOS virtual box created in Project 1.

Implementation

The main pwordcount file consisted of one function, main, which creates two pipes for communication between child and parent. The parent focused on reading in a user supplied word file, reading the file contents into a char array, then passing that array to the child. From here the child process would read in the file contents as a char array, count up all words inside by counting spaces, tabs and whitespace, and finally send it back to the parent as an integer. Helper functions are used from the helper file to count the words and supply error checking messages. Now the parent would read in the integer result and print to screen. A trace of how the pipes interact is printed to screen as can be seen in Diagram 1.

Diagram 1: pwordcount input and output



```
[root@localhost project2]# ./pwordcount input.txt

Process 1 is reading file "input.txt" now ...

Process 1 starts sending data to Process 2 ...

Process 2 finishes receiving data from Process 1 ...

Process 2 is counting words now ...

Process 2 is sending the result back to Process 1 ...

Process 1: The total number of words is 8.
[root@localhost project2]#
```

One last thing of note for the pwordcount file is that char array size is statically set to a max of 10,000 characters. This assumes the file will be no larger than that amount. This shortcut was said to be allowed in the case dynamic file processing was an issue. The helper function consisted of only two functions; a print function for error printing and a count function to count all whitespace in a char array. The make file consists of creating the main executable and deleting an existing executable for cleanup.

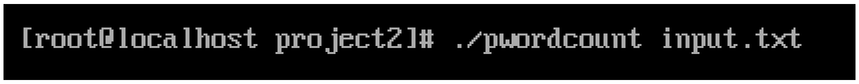
Testing

Unit testing was done on all helper functions and main functions. Several input files were also tested with various word lengths. Testing resulted in successful results.

Usage

To use the pwordcount tool, supply one input file below the max character length of 10,000 in the following manner:

Diagram 2: pwordtool usage



```
[root@localhost project2]# ./pwordcount input.txt
```

Results should include a trace of how the pipes interact and a resulting wordcount of supplied file. See Diagram 1.