Will May ([wdm0032@auburn.edu](mailto:wdm0032@auburn.edu))

COMP-3500-002

Homework 1

**Part 1:**

**1.)** Please open attached file “*HW1\_wdm0032\_COMP3500.cpp*” for source code.

Open “*input.*txt” for input file.

Open “*output\_part1.txt*” for output file.

(Compilation and execution instructions are included in source code file.)

**2.)**

**a.)** The parent/child process synchronization approach that was implemented in the program included the child process counting the number of characters in the input file and making the count available to the parent process. After this was done, the parent would use the count to keep track of the total number of characters as well as to output the line number, number of characters per line, and total number of characters to the output file.

**b.)** The inter-process communication (IPC) approach that was used to exchange data among the processes was the Shared Memory approach. This approach enabled the child process and parent process to both communicate indirectly and directly with the output file. The child process would send data to the parent process to output to the file, the parent process would also keep track of the processId, line number, and total character count to print to the output file. These two processes would therefor communicate their memory to the output file.

**c.)** A different IPC approach that could be used for this program could be pipes. Pipes would be available to communicate unidirectionally between the processes. The child process could use a pipe to write to the pipe, and the parent process could read the pipe and output the data to the output file. I did not use this approach because I did not understand the concept enough to implement it into my program, nor was any resource provided to sufficiently incorporate into my program.

**Part 2:**

**1.)** Please open attached file “*HW1\_wdm0032\_COMP3500\_part2.cpp*” for source code.

Open “*input.*txt” for input file.

Open “*output\_part2.txt*” for output file.

(Compilation and execution instructions are included in source code file.)

**2.)**

**a.)** The parent/child thread synchronization approach that was implemented in the program included the char\_Count method providing the data to the main method. The main method would go through each line from the input file and use the data from the char\_Count method to use to print to the output file.

**b.)** The approach that was used to exchange data among the processes was the pthread approach provided in the lecture “Threads” powerpoint. This approach would create a thread for each line and use the char\_Count method to output the line number, number of characters per line, and total number of characters to the output file.

**Part 3:**

**1.)** The implementations for Part 1 (process-based) and Part 2 (thread based) were two different approaches to reach the same end-result: to output the line number, number of characters per line, and total number of characters to the output file. The approach used in Part 1 used the fork() call in order to create a parent and child process. The approach used in Part 2 used pthreads to create different threads for each input line. After analyzing both approaches, it is desirable to use the thread approach for this program. Not only was this approach easier to implement, but it also allowed for better scalability and was more responsive. In terms of scalability, the thread approach was easier to handle multiple lines of input and multiple threads, while remaining in an organized structure. In terms of responsiveness, the thread approach executed the code quicker than the process approach, this could be due to the process/fork() approach requiring several if/else statements. Had the program only required one line of input file to be processed, the process approach could possibly be much more efficient and responsive.