COMP 3500 Homework #1

Maximum Points Possible: 100
Team Assignment (1..2 members per team)

There should be no collaboration among students (teams). A student/team shouldn't share any project code with any other student. Collaborations among students in any form will be treated as a serious violation of the University's academic integrity code.

Objectives: To learn the following.

1. Process management
   a. Creation of processes.
   b. Synchronization of parent/child processes.
   c. Inter-process communication approaches and applicability.
2. Thread management
   a. Creation of threads.
   b. Synchronization of threads.
3. Usage of POSIX PThread library.
4. Read/Write text files using a C Program.

Instructions:

1. This project can be submitted individually, or in teams of two members only.
2. Program must be written in C language.
3. Your program must take the input file name and output file name as command line parameters.
4. Assume that the input file only has alphabetic, numeric, and special characters.
5. Your program will be tested with multiple input files.
6. Make no assumptions regarding the length of input file.
7. Perform necessary error checking of command line parameters.
8. Perform appropriate error checking at each step.

Deliverables:

1. C/C++ file implementing Part-1 of the problem.
2. C/C++ file implementing Part-2 of the problem.
3. A sample input file; and
4. Output file from the test run
5. Execution instructions, if any

Part – 1 (45 Points)

1. (30 Points) Write a C program that counts the number of alphabetic characters from the given input file and prints the count, as follows.

(a) Read a line from input file.

(b) Create a new child process.

(c) Make the input line available to child process.

(d) Child process should scan through the input line and count the number of alphabetic characters ([a..z][A..Z]) in the given input line.

(e) Child process should make the count available to the parent process, and exit.

(f) Parent process keeps track of the following in a list:

   a. Process id of child process

   b. Input line number processed by the child process

   c. Count of alphabetic characters received from the child process.

(g) Repeat steps 1(a)...1(f) for each line from input file.

(h) Finally, parent process prints the following to output file

   a. (In increasing order of line numbers) Input line number, alphabetic character count.

   b. Total alphabetic character count in given input file.

2. (15 Points) Provide following:

(a) Describe the parent/child process synchronization approach implemented in the program.

(b) Describe the inter-process communication (IPC) approach you have selected to exchange data (both input and output) among parent and child processes. Provide reasoning for your selection.

(c) Briefly describe a different inter-process communication approach that can be used in this program. Provide reasoning for NOT implementing it in current program.

Part – 2 (45 Points)

1. (30 Points) Re-implement Part – 1(1): steps (a)...(h) with POSIX Pthreads as follows.

(a) Instead of creating a new process in step Part – 1(1)(b), create a new thread, and assign it the responsibility of processing the input line. Child thread should provide the alphabetic character count prior to exit.

2. (15 Points) Provide following:

(a) Describe the parent/child thread synchronization approach implemented in the program.

(b) Describe the approach you have selected to exchange data (both input and output) among parent and child threads. Provide reasoning for your selection.

Part – 3 (10 Points)

1. Compare and contrast the two implementations: Part – 1 (process-based) and Part – 2 (thread-based). Is one approach (always) preferable over other? Provide reasoning for your views.