CS5323 – Operating Systems II
Programming Assignment 1
Due: February 4, 2021, 11:59 p.m.
Submission: via Canvas

We have seen that computer hardware has been extremely cheap and hence have given rise to multiprogramming systems where each computer has multiple cores, with each executing independent streams of instructions such as different programs (e.g., running your web browser at the same time as your music player). A more complex, often underexplored use is to split up the work of a single program to speed up its completion.

In this assignment, you will be writing a simple C code to compute histograms of characters that leverages the power of multiprogramming. Your program should read a text file and print a histogram of letters present in each line. However, you must apply your knowledge of threads and processes to parallelize it to run faster.
There are two parts to this assignment. First, you must decide whether multi-threading or multi-processing is more appropriate for this use-case. Write a short report (1-page) choosing between the two and justifying your choice. Second, you must write a C program to read a file, compute the histogram and print it per line. Your code must take in as argument the number of threads/processes. Each thread/process must process one line and print its histogram and exit.

Make sure that your code is not data dependent i.e., each line must be processed by exactly one thread/process. The data to test your code will be provided.

An example of a histogram for a line is given below:

Input: *The quick brown fox jumps over the lazy dog*
Output: `{'a': 1, 'b': 1, 'c': 1, 'd': 1, 'e': 3, 'f': 1, 'g': 1, 'h': 2, 'i': 1, 'j': 1, 'k': 1, 'l': 1, 'm': 1, 'n': 1, 'o': 4, 'p': 1, 'q': 1, 'r': 2, 's': 1, 't': 2, 'u': 2, 'v': 1, 'w': 1, 'x': 1, 'y': 1, 'z': 1}`

Deliverables:

1. A short report describing your choice of multiprocessing vs multithreading and a brief justification. [10 points]
2. A well-documented code implementing the multiprogramming approach to creating histograms as described above. This includes comments, README file, instructions on compiling and running your code, etc. [40 points]

Things to remember:

1. The number of processes/threads entered on the command line must be greater than 2 and less than 4.
2. In case you decide to use multi-processing, be extremely careful that a child process does not itself fork a process or you can fill the process table and lock up the machine.
3. Submit your assignment as a ZIP file. Include a README with instructions on how to run and expected output along with your report.
4. **If you lock up another machine trying this assignment out, it is a 0 for this assignment!**