Assignment 1 - Multi-tasking and Synchronization

Written problems (Individual)

Problem 1
Consider the following processes arriving to the system:

<table>
<thead>
<tr>
<th>Process</th>
<th>Arrival time</th>
<th>Service time</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>P2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>P3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>P4</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>P5</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>P6</td>
<td>27</td>
<td>4</td>
</tr>
</tbody>
</table>

a) Calculate the average turn-around time and response time when using FCFS and SJF scheduling policies.

b) Calculate the average turn-around time and response time when using the Round Robin scheduling policy with a quantum of 4 seconds.

Problem 2
Consider the problem of servicing customers at McRonald’s, a very popular fast food chain in your neighborhood. McRonald’s is known for its delicious burgers at a fair price, and quick service!

The great service is due to the fact that cooks in the restaurant always pre-prepare the burgers, and customers can just pick up their burgers and check out. To be more specific, a customer goes into the restaurant, and heads to the service counter. If a burger is available, the customer picks it up, and then goes to the cashier to pay for it. If no burgers are available, the customer waits until the cooks prepares a new burger.

Cooks prepare burgers only if there is an available spot for them on the service counter. The service counter can hold up to 20 burgers, and if the service counter is full, the cook stops making burgers until a spot opens up.

Write the sequence of actions to be performed by a customer and a cook in McRonald’s. For simplicity, assume that there is only one type of burger sold at this branch. You can use as many semaphores as you need, but perfect grades will only be given to the logic using the minimum number of semaphores.
Let’s explore the power of distributing computation!

**Task 1**

Implement a single-threaded WordCount application, which takes as input a directory of text files, and generates a single output file with the frequency count of each word in all documents. Use it to analyze any of your favorite books.

**Example:**

Input document with text: “Be not afraid of greatness: some are born great, some achieve greatness, and some have greatness thrust upon them”.

Will generate an output file with the text:

Be: 1
Not: 1
Afraid: 1
Of: 1
Greatness: 3
Some: 3
Are: 1
Born: 1
Great: 1
Achieve: 1
And: 1
Have: 1
Thrust: 1
Upon: 1
Them: 1
Task 2
Implement a multi-threaded WordCount application.

In this application, you will create multiple threads to speed up the computation of WordCount from task 1 above. A single thread takes as input one or more text files, and generates the frequency count of each word in the documents. You can synchronize the threads in any way that you see fit, but note the following challenges:

1. You will need to divide the data among the threads somehow.
2. You don’t want two or more threads to process the same document (or part of document).
3. All output needs to be placed in a single output file at the end of the program.

You can use as many data structures and as many Semaphores, as you see fit, in this task.

Step 3
Write a report on your work on task 2. In this report you will need to include the following:

- How did you solve each challenge in task 2? Be specific, add pseudo code if you need to.
- Did you face any other challenges? How did you solve them?
- Compare the running time of your code in task 1 to that of thread 2 with a single thread. Explain your results.
- Calculate the running time of task 2, using a varying number of threads. Start with a single thread, and keep increasing the number of threads till the running time doesn’t change (significantly).
- Create a plot to represent the varying running time with increased number of threads. Explain your results.

Deliverables
- The code, of course!
- The report from task 3.
- A document summarizing the group discussions, prior to the implementation. It should include:
  - The main ideas discussed within the group
  - The references/readings used during the discussion
  - The challenges faced while discussing an appropriate solution