

PA4: write a multi-threaded program using pthreads CMPSC 473, FALL 2017

Released on Nov. 6, 2017, due on Nov. 19, 2017, 11:59:59pm

Bhuvan Urgaonkar

Disclaimer: Although the characters in this story are based on historical figures, the events are based purely on the author's imagination.

1 Problem Description

It is 1783 and Robert Morris and some of his rich friends are sending *The Empress of China* on what will become the first voyage of a commercial ship from the newly independent USA to the empire of China. The ship's captain John Green, a former US naval officer, must be careful about solving a synchronization problem to ensure the voyage is successful. He must make a stop at Calicut, a port in southern India, to wait for three East India Company merchants to bring in opium grown in the province of Bihar in India, which will become part of his ship's cargo. These merchants are unpredictable and so Capt. Green must be prepared to deal with uncertainties. If, upon reaching Calicut, he finds that one or more of these merchants have already sold their goods and left, he must turn back and return to the USA. Else, he must wait for them to arrive (at uncertain times in the future) before he moves on to Canton.

Imagining The Empress and the three merchants to be threads within the same process, write code that accomplishes what Capt. Green desires.

2 Code, Inputs, Outputs

You will use the `pthread` library to write your program. Your program will take four inputs representing the arrival times (each expressed as a day in the range [1,10], the time-of-day does not matter) of The Empress and the three merchants, respectively. You will find several files, each specifying one such set of inputs, in the GitHub repository for PA4. The main thread should launch the 4 threads mentioned above and then wait for them to finish. It should then output relevant events and details of the decision-making and scheduling done by your synchronization code. Following this, it should conclude the program.

You may choose your own output format but it must offer clear details for the TAs to check the correct working of your program. You will use mutex locks and condition variables for ensuring safety. Your code must be deadlock-free.

3 Submission and Grading

PA4 will be done in groups of 2. A group may choose either member's repository as their workplace. Do remember to list both members names in all files you submit (including your report). Your submission won't be graded if you don't have a name on your report. You are supposed to submit your assignment with your private GitHub repository. Therefore, accept the invitation to access your private repository. If you still need instructions for working with git or adding your teammate to your repository, refer to the included Git manual file. The TAs will grade you by inspecting your code, running some test cases for your code, and by looking at your report.

4 Submission

You must push and commit to your private repository all your source files (*.c and *.h), your report, and a Makefile that the TAs will use to compile your code. You should also create a short README with instructions on how to compile and run your code. Your report should contain: (i) pseudo-code for your solution (in the style of code written in class) and (ii) outputs for all the input files. The TAs may additionally use some undisclosed inputs to check the correct working of your code.

5 Some final remarks

- **IMPORTANT: We will be comparing your codes for similarity and all parties involved in copying will receive a 0 grade on PA3 at the very least and an F grade in the course in the worst case. Outsourcing your code to an external entity is considered cheating. You will also be reported to the college of engineering disciplinary committee.**
- Please seek as much as help as you need from the instructor and the TAs.
- Good luck and happy coding!