CS 347M Operating Systems Minor
Programming Assignment No. 2: Concurrent Programming

13 March 2017

You have to develop a concurrent program **using pthreads** to implement an airline reservations system subject to the following specification:

1. The airline reservations system consists of a **master thread** and a **pool of 5 slave threads**, where each slave thread can perform the following operations for a passenger: Check availability of a seat on a specific flight (operation *status*), Book a seat on a specific flight (operation *book*), and Cancel a booking (operation *cancel*). **You can decide how the threads are named (but they must have distinct names!), and what parameters the operations should have.**

2. A slave thread is blocked until the main thread assigns a query to it. When a query is assigned to it, it becomes active, processes the query and blocks itself after completing processing of the query.

3. The system contains a database containing details of 10 flights, and the booking status of each of these flights. The consistency of this data base is to be safeguarded against wrong operations and race conditions. **Again, you can decide how the data is actually stored, and what their initial values are.**

4. The master thread reads details of newly arriving queries from a file named **transactions**. **You may assume your own format for the queries.** The last query in the file is followed by a string "END". Operation of the system completes when all queries have been processed. The master thread should print a message saying that all queries have been processed.

5. When the master thread reads the next query from file **transactions**, it assigns the query to a slave thread if a slave thread is blocked. If none of the slave threads is blocked, it waits until a slave thread becomes blocked and assigns the query to the slave thread and activates the slave thread. When it assigns the query to a slave thread, it transfers the details of the query to the concerned slave thread and prints a message containing the query and the name of the thread to which the query is assigned. It reads the next query only when it has assigned

the previous query to a slave thread. **You must set up your own arrangement for passing details of a query to a slave thread.**

6. A slave thread reads details of the query handed to it, carries out the appropriate actions such as updating the database, and prints a message containing the query, its own name, and the result of the query. After performing these actions, the slave thread blocks itself waiting for the master thread to assign it another query.

7. You must ensure (a) correctness of query processing, (b) data consistency, (c) freedom from busy waits, and (d) maximum concurrency in the processing of queries.

Implementation notes:

1. Remember that the master thread and the slave threads share the code and data spaces. **However, you must ensure consistency of the data.**

2. First think of the synchronization requirements of the threads, then think of how you would implement each of the requirements, and only then begin coding.

Readying a demo

For evaluation of the assignment, you will be asked to demonstrate working of the system and create various situations of interest, for example, a situation in which some of the slave threads are blocked while others are active, the situation in which all slave threads are blocked, when the last query is being processed, etc. So you should identify such situations, and be able to create them by creating appropriate transaction files and/or adjusting the waits in the slave threads.

What should you submit

Submit a README file and your code.

Submission deadline

8 am on Monday 4 April 2017.