CSCE 410 611 Project #1
Multiple Producer-Consumer Problem
Changed source code filename requirement to use a .c extension

Project Points: CSCE 410: 125, CSCE 611: 100

Deadline: NLT 2359 hours (11:59 pm) on ~~Thursday~~ Friday 25 February

Early turn-in credit ends on 2359 hours (11:59 pm) on Thursday 17 February
2 points per 24 hours before 11:59 17 February

This assignment requires an electronic submission of a source code file and a compiled executable.

Project Hard Copy Requirements:
There is no hard copy submission requirement for this project.

Electronic Copy Requirements:
Upload your hard copy source code in plain text format and use the following naming convention: name your file with last name and first initial underscore assignment name and appropriate extension for c source code files. I would use: the following name hamiltonj_proj1.c. If you prefer to use multiple source code files then make sure each source file ends with an extension of .c. Use the tar command on the Texas Cyber Range to create a tar file following the assignment naming convention with an extension .tar for me it would be hamiltonj_proj1.tar. Upload your source code to your account in the Texas Cyber Range. Compile your source code using gcc in your VM. Upload your correctly running code to the assignment page and name your compiled binary with last name and first initial underscore assignment with an ".exe" extension. (Unix-based systems execute files based on the magic number, not the file extension, the use of ".exe" is simply as an aid to grading.) I would name my binary file hamiltonj_proj2.exe.

Mandatory use of the Texas Cyber Range:
You are free to write your program on any platform you wish, BUT you must compile your code using gcc in your virtual machine on the Texas Cyber Range.

Problem Statement:
Develop a "correct solution" to the multiple dimensioned, multiple producer/consumer problem. (That is multiple producers > 3 and multiple consumers > 3) problem that send and receive messages directly among themselves. Make sure that each message is received by one and only one consumer and that every producer and consumer is treated in a fair manner (no starvation). No centralized task manager is allowed. That is, you cannot simply queue all requests from each producer and then sequentially assign them to consumers. *No common shared memory*.
Producer:
1: if msg from Consumer, place msg on queue w/timestamp

2: send ack to Consumer (I.e., I have a resource)
Consumer:
1: determines resource need.
2: receives resource and processes it, return to step 1
To demonstrate your solution, you must illustrate it with actual running code (that completely documents your solution), and examples. (I.e., a correct implementation of Presser, Computing Surveys, 7,1, March 1975, pp.21-44).
Use gcc on your Texas Cyber Range VM to implement your solution.

Addendum:
This project is an **individual project.** Each student is required to implement his or her own solution. Discussing solutions with others in the class is OK – copying from other students is not. Two requirements here: If you discuss solutions with others, (and I encourage you to do so) list your study partners in the header of your solution. Recall also that I expect original work and so a project that details how it was written by half the class will not be well received. However, I respect good scholarship and academic integrity – keep that in mind as you document your projects.

You may **not** copy solutions from other members of the class.
Bottom Line: Academic integrity and good scholarship are paramount. If your project implementation is not original and not documented, then we potentially have a case of academic dishonesty (cheating) and that will be pursued in accordance with university procedures.

Always feel free to ask questions regarding academic integrity.