

CMPT 300 – Operating Systems

Fall, 2017

Due: Friday, October 27th, 2017.

HW2: Bounded Buffer Problem

PROBLEM STATEMENT:

There are 3 generators and each produces a unique kind of material independently. All these materials are stored in an input buffer with size 10 before they are forwarded to the operators. We have 3 operators with same priority who are responsible for producing the products based on these materials. Each product needs 2 different kinds of materials. Each time an operator needs 2 tools for this purpose. There are totally 3 tools provided for these operators. An operator can only process one product at one time. When an operator gets both the materials and tools, he can produce a product within a limited time varied from 0.01 second to 1 second. Otherwise, he has to wait until all the necessities are met. He can grab the materials or tools first, it does not matter, but he can only get one thing at one time. If an operator decides to make another product before he starts to make the current product, he can put the materials and tools back and re-get the new materials and tools. But he has to put the tools back after he finishes a product because other operators may need these tools. All the products are put into a size-unlimited output queue. An operator cannot start a new product before he puts the product into the output queue. Some restrictions may apply to these products: *1)* No same products can be next to each other in this queue. We say that two products are same if they are made from the same kinds of materials. *2)* The difference of the number of any two kinds of products produced should be less than 10, for example, we can have 10 of product A and 15 of product B, but it is not allowed if we have 10 of A and 21 of B because the difference is 11 which is larger than 10.

SUBMISSION INSTRUCTIONS

Your assignments must be both emailed and presented in class on the due date. You will submit a single .ZIP, .RAR or .TAR file that will contain:

- 1) A soft copy of your source code.
- 2) A **Makefile** to compile the code.
- 3) A soft copy of a report documenting the internal design of your program.

Your program must be in C/C++. It is your responsibility to make sure your program compiles and runs smoothly under Linux. Your program should not be stalled in a dead cycle at any time. This is the only submission method

.

DEADLINE

The deadline for homework is 11:59:59 PM on **Friday, October 20th, 2017**. Submissions after this deadline will be considered late and will NOT be accepted. Submission details will be provided by your TA.

GRADING

Your homework assignment is worth 10 points (10%). During the execution of your program, you should provide these information dynamically:

- 1) For each material, how many of them are generated?
- 2) The status of the input buffer.
- 3) For each kind of product, how many are produced?
- 4) The status of the output queue.
- 5) How many times the deadlock happens?

Additional Notes:

Students need to be careful to design their solution in a way that avoids any potential deadlocks that could be caused by these scenarios:

- A glut of materials of the same type in the materials queue (solution: students should incorporate some mechanism to guarantee that there is sufficient variety in the materials queue)
- Workers hogging tools (Possible solutions: workers attempt to grab PAIRS of tools instead of picking them up individually. Alternatively, when a worker is holding on to a single tool, there needs to be some mechanism like a counter that keeps track of attempts to grab the second tool. After a certain number of unsuccessful attempts, the worker should release the tool its holding, which would allow some other thread to grab it.)

Also provide these functions

1) Pause and resume the program at any time. 2) Make the number of operators and tools adjustable.

Please feel free to provide additional information and functions that you consider useful.

The most important thing is to make sure your program works correctly. Do not worry about your programming style or efficiency. But it does matter if you provide a smarter internal design (approach) and user-friendly interface.

IMPORTANT: Your program must COMPILE AND RUN or you will get a mark of ZERO on your assignment.

Assignment grades will be open for review for one week only.

Finally, a reminder that plagiarism will result in an automatic failing grade for the course.