Simulation Lab 2: Process-Consumer Simulator

Submission to the bb4.utc.edu is due on midnight of XXX.

Name your submission file with firstName_lastName_Simulation2.

Turn In:

Include your experiences and answers to Part I (question 1 to 5) and Part II (question 1 to 5).

Include the following at the beginning of your report.

- Name: ____
- UTC ID: ____
- Course Number and Name: _____________________________________
- Semester: ____
- Lab Name and Number: ____
- I spent ____________hours and __________minutes to finish this hands-on lab.
- I have _____________ (percent) finish this lab.
- I expect ___________ (A, B, C, or F) of this lab.
- This lab helps me to understand process scheduling. Choose a number to indicate how much the lab is helpful.
  1 2 3 4 5
  (less helpful) (more helpful)

Part I
Instructions: Download the Producer-Consumer simulator and extract the zip file. This produces a folder named **pc**. The user manual for this simulator, **pc_doc.html**, is included in this folder. It is strongly suggested that you carefully read through this documentation describing how the simulator operates prior to beginning this section.

Steps: Perform the following steps using the producer-consumer simulator:

1. In the **pc** directory, execute `./runpc` (UNIX, Linux, Mac OS X) or `runpc.bat`(Windows). This will start the producer-consumer simulator.

2. Click on the green button labeled **One Run** in the right-hand side of the Producer Consumer Simulation window. This will start the simulator for a single run -- the simulator stops upon either an internal or external failure condition.

3. Click the **Reset** button in the bottom row of buttons, in the second column from the left, to reset the simulator. At the bottom right-hand side of the window is the number of cycles remaining in the quantum (appears as **Quantum Remaining: 100**). Click the

green button labeled **10 Cycles**. Click the **10 Cycles** again. (Do not click the 10 Cycles button more than twice.) The value of Quantum Remaining is decrementing and the code statements in the producer are being performed.

4. Click the green button labeled **One Cycle** three times. The value of Quantum Remaining is not changing, yet code statements in the producer are still being performed. That is, by clicking the **One Cycle** button, each cycle is being used to execute code statements, yet the cycles do not count towards the time quantum of the scheduled process.

5. Click the **One Run** button again to run the simulator until a failure condition occurs.

6. Exit the simulator using the pink **Quit** button in the upper right corner of the main window.

7. Edit the file `pcconfig` in the **pc** folder and change the value of **user** from **New User** to your name. Save the configuration file.

8. Execute the simulator again (as in Step 1).

9. Click the **Open Log** button in the left column, about two-thirds down the Process Consumer Simulation window to open a log file. (After you have clicked this button, its name will change to **Close Log**.)

10. The button beginning with the label **Failure Type:** in the second column from the left, toward the bottom of the window, allows you to toggle between an **Internal Inconsistency** failure or **External Inconsistency** failure. If necessary, click this button so that it is an **Internal Inconsistency** failure.

11. Now click the green button labeled **One Run** at the right-hand side of the window.

12. Click on the green button labeled **Show Local Log** at the left-hand side of the window. A table opens up in a separate window with statistics for the simulation. At the bottom of the log is a table with statistics for each line of code executed by the producer and the consumer. Appearing in red text is the internal inconsistency failure with a line such as the following:

    Items Produced - Items Consumed = 2 != 3 = Number Buffered

13. Return to the Producer Consumer Simulation window and select the **Failure Type** button so that it is an **External Inconsistency** failure. Click the green button labeled **10 Runs**.

14. Click on the button labeled **Show Local Log**. A table opens up in a separate window with statistics for each of the two simulations performed at Steps 11 and 13.

This document was derived from simulation software created by Steve Robbins which was supported by NSF DUE-9752165

15. Return to the Producer Consumer Simulation window and click the **Close Log** button, then the pink **Quit** button. You have created a file called **log.html** in the **pc** directory.

16. Submit your log file -- **log.html** -- to your instructor per his or her instructions.

Questions:

Answer the following questions after completing Steps 1-16 above.

1. What process is initially active after completing Step 1?
a) Producer
b) Consumer

2. The purple button labeled **Quantum** is followed by a number specifying how many cycles in a quantum. What is this value?
a) 10
b) 100
c) 90
d) 5000000

3. Next to each instruction for the Producer and Consumer processes is the number of cycles required by that instruction. How many cycles is required for the statement "**`while (counter ==n)`**"?
a) 1
b) 7
c) 8
d) 10

4. What is the value of Quantum Remaining in the lower right-hand corner of the main window after performing Step 3?
a) 100
b) 99
c) 0
d) 80

5. What is the value of Quantum Remaining in the lower right-hand corner of the main window after performing Step 4?
a) 80
b) 79
c) 78
d) 100

Part II

Instructions: If you have not already done so, download the Producer-Consumer simulator and extract the zip file. This produces a folder named **pc**. The user manual for this simulator, **pc_doc.html**, is included in this folder. It is strongly suggested that you carefully read through this documentation describing how the simulator operates. It is also recommended that you complete Part I as well.

The following activities allow you to manually step through the producer and consumer processes to ensure a context switch occurs while the producer is executing its critical section. You will perform this by using a combination of the **10 Cycles** and **One Cycle** buttons that allow you to step through code instructions in the producer and consumer processes. Initially, you will use these buttons to have the producer run its code segment almost two times, stopping while in the middle of its critical section for the second time.

Steps:

1. In the **pc** directory, execute `./runpc` (UNIX, Linux, Mac OS X) or `runpc.bat` (Windows). This will start the producer-consumer simulator.
2. Click the **10 Cycles** button several times so that the producer complete its first time quantum. The simulator will switch from the producer being active to the consumer.
3. Next, click the **Switch Processes** button to switch back to the producer process.
4. Using a combination of the **10 Cycles** and **One Cycle** buttons, ensure that the producer has entered one item in the buffer and proceed to the instruction `R1 = counter` for the second time.
5. Click **One Cycle** to step to the `R1 = R1 + 1` instruction.
6. Click the **Switch Processes** button in the fourth column at the bottom of the window to switch to the consumer process. This has the effect of moving the producer process off of the CPU while it is in the middle of its critical section.
7. Using a combination of the **10 Cycles** and **One Cycle** buttons, allow the consumer to proceed at least until it decrements the value of `counter`. (You may allow it to run for the remainder of its time quantum or until it yields control of the CPU to the producer.)
8. Switch back to the producer process and click the **One Cycle** button twice so that it completes its critical section. You have now ensured a race condition on the shared variable `counter`. (It may take you a few attempts at the above steps to cause a race condition on the data `counter`.)

1. What is the value of `counter` after completing Step 4?
a) 0
b) 1
c) 2
d) 3

2. What is the value of **`counter`** after completing Step 7?
a) 0
b) 1
c) 2
d) 3

3. What is the value of **`counter`** after completing Step 8?
a) 0
b) 1
c) 2
d) 3

4. What is the correct value of **`counter`** after completing Step 8?
a) 0
b) 1
c) 2
d) 3

5. Identify the type of failure that has occurred.
a) Internal inconsistency failure
b) External inconsistency failure