

**CMPSC 440**  
**Operating Systems**  
**Spring 2014**

**Laboratory Assignment Three: Using a Multi-Threaded Producer-Consumer Model**

## Introduction

Processes and threads are commonly used by the operating system itself and by the programs that run on the operating system. In this laboratory assignment, you will download, use, extend, and experiment with a multi-threaded producer-consumer model. The producer-consumer model, sometimes known as a bounded-buffer model, illustrates non-deterministic execution of threads that share a common, fixed-size buffer. In this assignment, you will download and use a working version of this model. Then, you will add some features to the model that enable you to experiment with it more effectively. Finally, you will systematically break various parts of the model in order to observe what can go wrong when implementing multi-threaded Java programs.

## Accessing the Producer-Consumer Model

During this laboratory assignment, and some subsequent assignments, we will securely communicate with the Bitbucket.org servers that host the source code for our assignment. Students who are already comfortable with using Git and Bitbucket may skip the majority of these steps and simply ask the instructor to share the course repository with them. For those students who are not yet using Git and Bitbucket, this laboratory assignment will teach you all of the steps needed to configure the accounts on the departmental servers and the Bitbucket service. Throughout the assignment, you should refer to the following Web site for additional information about the use of Git and Bitbucket: <https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101>.

1. If you have never done so before, you must use the `ssh-keygen` program to create secure-shell keys that you can use to support your communication with the Bitbucket servers. Type `man ssh-keygen` and talk with the instructor to learn more about how to use this program. What files does `ssh-keygen` produce? Where does this program store these files?
2. If you do not already have a Bitbucket account, please go to the Bitbucket Web site and create one — make sure that you use your `allegheny.edu` email address so that you can create an unlimited number of free Bitbucket repositories. Then, upload your ssh key to Bitbucket.
3. Now, you need to test to see if you can authenticate with the Bitbucket servers. First, show the course instructor that you have correctly configured your Bitbucket account. Now, ask the instructor to share the course's Git repository with you. Open a terminal window on your workstation and change into the directory where you will store your files for this course. For instance, you might have made a `cs440S2014/` directory that will contain the Git repository that I will always use to share files with you. Once you have done so, and only after I have shared the Git repository with you, please type the following command: `git clone git@bitbucket.org:gkapfham/cs440s2014-share.git`. If everything worked correctly, you should be able to download all of the files that you will need to use for this

laboratory assignment. Please resolve any problems that you encountered by first reviewing the Bitbucket documentation and then discussing the challenges with the Git experts in this class. If you are still not able to run the `git clone` command, then please see the instructor.

4. Using your terminal window, you should browse the files that are in this Git repository. In particular, please look in the `labs/lab3/src/` directory and use Vim to study the Java programs that you find. What files are used to implement the producer-consumer model?

## Understanding and Extending the Producer-Consumer Model

After you have carefully studied the source code of the producer-consumer model, you can compile and execute it. What type of output does this program produce? Will this program halt? If yes, then how long will it take to halt? If no, then why does it not halt? Finally, you should use the `/usr/bin/time` program to time how long it takes to run the `ProducerConsumer`.

In order to better understanding how this program works, you should run it and then use the `ps -eLf` and `ps aux` commands to track what Java processes are started when the model executes. You can also use the `gnome-system-monitor` to learn more about this program's behavior (to run this program you can type `monitor` in the Unity dash). To the best of your ability, you should use these three programs — and any others that you deem to be useful and relevant — to better understand how the `ProducerConsumerModel` creates processes and/or threads.

As you will see from studying the source code, the current implementation of the `ProducerConsumerModel` has several hard-coded variables that control its behavior. Leveraging the `JCommander` system that you learned how to use in a previous laboratory assignment, you should now implement a command-line interface for the `ProducerConsumerModel`. This interface should make it possible to specify whether or not debugging output must be produced, the number of data items that must be produced and consumed, and the total number of consumers. Details about the command-line arguments are provided in the comments of the `ProducerConsumerModel.java` file.

## Understanding Defects in Multi-Threaded Programs

After you have finished writing the source code to support the command-line arguments for the `ProducerConsumerModel`, you can take steps to systematically explore what happens if you make mistakes when implementing multi-threaded Java programs. Please ensure that you carefully make the following changes, recompile the program, execute the program, record the incorrect program output, and comment on why the output is evident. Once you are done making the required defective implementations of the program, please return the modified code to the correct state!

1. For the `get` and `put` methods inside of the `CubbyHole`, systematically add a new method signature that does not declare the method to be `synchronized`. That is, you should first run the `ProducerConsumerModel` with just the incorrect `get`, then with just the incorrect `put`, and finally with both the wrong `get` and `put`. What does the output of the defective `ProducerConsumerModel` look like? How can you explain this output?
2. Now, keep the version of the `get` and `put` methods that do not use the `synchronized` keyword and separately comment out the `while` loop in both of the `get` and `put` methods. Finally,

you should separately comment out the call to the `notifyAll` method at the end of both of these methods. What does the output of each of these `ProducerConsumerModels` look like? Can you explain why these defects produce the output that they do?

## Experimental Study of Performance

Now, you should use the correct implementation of the `ProducerConsumerModel` to conduct an experiment. You should measure the time overhead that is required to consume 10, 100, 1000, and 10,000 different data items for three different numbers of consumers. For example, you might use two, four, and eight different consumer threads for all of the different number of data items. You should conduct each individual experiment at least five times in order to facilitate the calculation of arithmetic means. After making sure that your timings do not include the cost associated with performing console output, you should collect execution timings of the `ProducerConsumerModel`.

In preparation for writing your report, you may calculate arithmetic means by implementing your own Java program or by using the R language for statistical computing. Moreover, your report should explain your experimental goals and design by clearly describing the commands that you type and the order in which they were typed. Your report should also include tables of results that furnish the running times for each of the different `ProducerConsumerModel` configurations.

## Summary of the Required Deliverables

This assignment invites you to submit printed and signed versions of the following deliverables:

1. A paragraph that explains how multi-threaded Java programs use `synchronized`
2. A detailed discussion of the execution behavior of a multi-threaded Java program
3. The output from running the defective versions of the producer-consumer model
4. A comprehensive analysis of the output of each defective multi-threaded Java program
5. The source code of all of the classes in the enhanced producer-consumer model
6. The report from an experimental study that characterizes the performance of the model
7. A reflective discussion of the challenges you encountered during this assignment

Students are strongly encouraged to write their laboratory report in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  and use tools such as the R language for statistical computation to better analyze and visualize the empirical results. Please see the instructor if you have questions about the requirements for these deliverables.

In adherence to the honor code, students should complete this assignment on an individual basis. While it is appropriate for students in this class to have high-level conversations about the assignment, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else's work. As such, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code.