**Midwestern State University**

**Advanced Operating Systems**

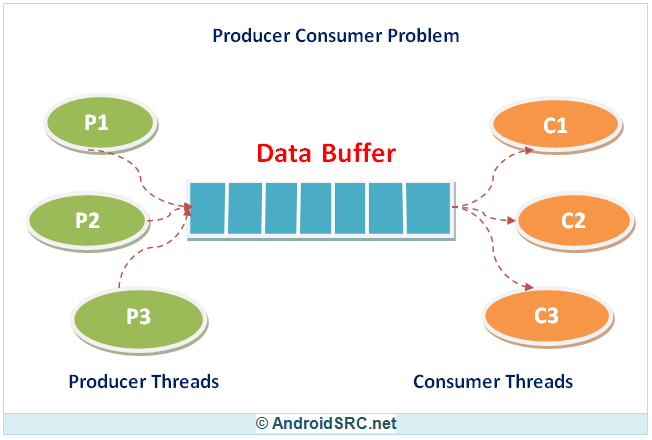
**Spring 2018 – Reading Quiz – 04/06/2018**

The objective of this quiz is to ensure that the students read/approach the material at the designated class time. In order to answer these questions, you need to read and summarize material from your book.

You are not allowed to copy info from the web, you are not allowed to write exactly the same info that the book has. I you are caught in such behavior, the rules of cheating and plagiarism will be applied. You are required to read and the *answer these questions in your own words.*

**Type here: Font: Times New Roman, Size 11, Spacing 1.0**

Samuel Weems

[[1]](#footnote-1)

1. What is the consumer producer problem (page 222)? Summarize this on your own words.

*You have 50 minutes to write your summary and upload it in D2L. If your summary is not delivered on time, then it will not be accepted.*

In this problem there are consumers and producers. Producers are defined as those that are generating or creating data. Consumers are defined as those that process the created data. As producers create new data it is placed in a buffer and the consumer pulls new data from that same buffer.

An associated problem with this setup is preventing producers from attempting to place items/data into an already full buffer and preventing consumers from attempting to remove items/data from an empty buffer. In order to prevent this problem from occurring the buffer must be protected so that only one consumer or producer can access it at any one time.

There are several methods or implementations to protect the buffer but in our case semaphores are examined. The first method explained by the book is the use of binary semaphores. While using two binary semaphores at first glance appears to be a successful method, there is the possibility of scenarios in which a type of race condition occurs because a producer increments a semaphore before a consumer can test it which then leads to a deadlock situation. The solution offered is to use an “auxiliary variable”. Another method discussed with a “cleaner solution” is the use of counting semaphores.

**Read, understand and be prepared to answer questions about the requirements for mutual exclusion (pag 212)**

1. http://androidsrc.net/producer-consumer-problem-java-detailed-explaination/ [↑](#footnote-ref-1)