

**Operating System (CSC 433A)**  
Computer Science Department  
**State University of New York at Plattsburgh**  
**PROJECT #2**  
**Producer - Consumer**

This project will be to implement the proper synchronization between Producer and Consumer threads using Semaphore and Lock.

Python has Semaphore and Lock which must be used to complete the implementation (<https://docs.python.org/2/library/threading.html>)

You must optimize your code, to minimize the duration at which the Lock is held by any of the thread.

The program has Producer and Consumer threads, which currently do not synchronize themselves between each thread of the same type and also between Consumer and Producer.

The program processes a file(function *loadTransaction*, which contain 3 fields per line: <transaction id>,<sleep time in ms for the consumer>,<sleep time in ms for the producer>, and load them in the list *transactions*.

The transaction must be processed in a FIFO manner between the producer and the consumer.

When a producer threads get a transaction from *transactions*, puts it in *fifo*, then sleeps the number of producer ms specified in the transaction.

When a consumer threads gets a transaction from *fifo*, it then sleep the number of consumer ms specified in the transaction before trying to obtain the next transaction from *fifo*.

What you must do:

1. Synchronize between the producers and the consumers, to have the producer stop when the FIFO is full (based on the *maxFifoDepth*), or to have the consumers stops when the FIFO is empty
2. Synchronize between the consumers and the producers, to have the consumer threads stop when the end of file is reached
3. Transaction.txt: there is one transaction file as part of the assignment. You must create additional files that you will use to test your implementation under multiple scenarios to prove that the synchronization is correct (1 producer+N consumers, N producers+1 consumer, N producers+N consumers, changes in the sleep, ....) and provide the documentation regarding each of these tests

Grade:

- synchronization between producers: 25
- synchronization between producers and consumers: 25
- synchronization on the end of file between producers: 25
- test files and parameters: 15

- documentation of the tests: 10
- Bonus: synchronization on the end of file between producers and consumers: 10

You must use Semaphore (to track the number of transaction in *fifo*) and Lock (for the critical section) see: <https://docs.python.org/2/library/threading.html>.

See textbook p. 130 Solving the Producer-Consumer Problem using Semaphores

### **Submission:**

One zip file per team, with the python implementation, the transaction files used to test your implementation(what were you testing) and a documentation(pdf) of the command line(arguments) used to test your program.

For each test you did, provide the command line and the output of the program.

Examples of test documentation:

Test # - Input filename - # of Producer - # of Consumer - Max FIFO Depth

1 - transaction.txt - 1 - 1 - 1

2 - transaction.txt - 1 - 2 - 5

### **Implementation Files:**

There is a startup implementation in python: project2.py and also an example of a transaction file: transaction.txt

Command line : project2.py <# of producer> <# of consumer> <maximum depth of the fifo>

As an example, to execute the program:

python project2.py transaction.txt 1 1 5