**Operating Systems**

**Final Project Part 1**

The final project consists of modifications to the producer-consumer server. For part 2, you will make a change to the server.

Currently, a producer thread in the server collects an item (the size and the buffer of letters) from the producer client, and then it places that item in the item buffer to be picked up by a consumer thread. After successfully placing the item in the buffer, it releases the producer client, by sending the message DONE.

In thus version, you will change the order of events, without changing the networking protocol. However, you will need to change the definition of an item:

typedef item\_struct {

uint32\_t size;

int prod\_sd;

}ITEM;

Now this item, rather than holding the data itself, tells the consumer where it can get the data (from the producer socket). The new sequence of events goes like this:

* A producer thread sends the GO message but reads and puts only the size (length) of the data into the item struct, along with the producer client’s socket id.
* It then waits (if necessary) to place that item into the item buffer.
* Once the item is successfully placed into the buffer, the producer thread is finished, but the socket is not closed (because the data has not been read).
* When a consumer thread picks up an item, it finds out the socket id of the producer client who can send it the data
* The consumer thread must read that data from the producer client and send it on to the consumer client.
* Once the consumer has received the all the data from the producer client socket, it can send the message DONE to release the producer client and close that socket.
* Once the consumer client has been sent all the data, that socket can be closed, and the consumer thread is finished too.

The size or length of the data is now specified precisely as a 32-bit unsigned int, to be precise.