Name: Garrett Bearss

Documentation

The producer and consumer processes use shared memory to share the values of the table. The producer writes values into the table (which holds 2 values) while the consumer takes in those values and consumes them from the shared memory, with only one process accessing the table of items at a time. The processes are controlled by semaphores to ensure the producer does not access the table while the consumer is, and vice versa. The producer is ensured that it can't write into a full table and the consumer can't consume from an empty table. If the producer or consumer are unable to perform their corresponding action, they sleep for a random set amount of time while the other process can perform their action. The processes also use a mutex to ensure mutual exclusion between both programs while they enter the critical section. After a process is done with the critical section, they will signal the other process to allow them to perform their action using semaphores. The process will continue for a defined number of items to be produced and consumer based on the variable loop in both programs. These variables must be equivalent to ensure the program works correctly.

The following commands are used to run the programs:

```
$ gcc producer.c -pthread -lrt -o producer
```

```
$ gcc consumer.c -pthread -lrt -o consumer
```

```
$ ./producer & ./consumer &
```