Authored by: Adam Bayley and Samantha Hawco

ELEC 377 Lab 3 Testing Documentation

Testing of Mutexes

After adding functionality to the provided function definitions of `getmutex` and `releasemutex`, the program was compiled through the vm to ensure it does not include any compiler errors. This was done to prevent larger errors during later testing.

Testing of Consumer and Producer

After being written using the provided pseudocode, the producer and consumer classes were built using the `make` command. Small errors such as incorrect casing and missing brackets were identified by the compiler and were then corrected. One error that was difficult for the group to correct was that `getChar()` was written when the correct function is `getchar()`. With the aid of the lab teaching assistant, this was corrected. After fixing these errors through repetitive edits and builds, the code compiled successfully.

To validate the functionality of the consumer and producer programs, the input from the producer program must be compared to the output from the consumer program. The input for the producers can be viewed in the documents `text00in.txt` and `text01in.txt`, while the output of the consumers can be seen in the document `text00out.txt`; both found in the lab3 git repository. The first text file contains letters “abcdefg” and the second file contains numbers “12345”. This reason for one file containing letters and the other containing numbers is that it makes it easy to interpret the consumer output and assess the functionality of the programs. To test, two producer programs and one consumer program were run in separate consoles so that they could run simultaneously and allow for testing of synchronous behaviour. The first producer was given the alphabetical file, `text00in.txt`, and the second producer was run with the numerical file, `text01in.txt`. The producers should be adding characters from their files to the shared buffer once space is available. The expected order of this is “abcdefg12345”. Next, a consumer was run and directed to write to `text00out.txt`. This file should contain the above order of characters to demonstrate the correct functionality and synchronous behaviour of the consumer and producer programs. The output shown in this file is the same as the expected output, therefore; indicates that the producer and consumer programs operate and use the shared buffer correctly.