CS 332011 - Operating Systems - Project 1

Generated by Doxygen 1.9.1

1 Programming Assignment 1 - Producer & Consumer Problem 1

1.0.1 Author: Elliott Hager 1

1.0.2 Fall 2023 Semester 1

1.0.3 CS-33211: Operating Systems 1

1.0.4 Dr. Qiang Guan 1

1.0.5 Kent State University 1

1.0.5.1 Description (Provided from the assignment instructions) 1

1.0.6 Implementation 2

1.0.7 Compilation 2

1.0.7.1 Consumer: 2

1.0.8 Producer: 2

1.0.9 Clean: 2

1.0.10 Run Instructions: 2

1.0.11 Libraries & Tech Stack 3

Chapter 1

Programming Assignment 1 - Producer & Consumer Problem

1.0.1 Author: Elliott Hager

1.0.2 Fall 2023 Semester

1.0.3 CS-33211: Operating Systems

1.0.4 Dr. Qiang Guan

1.0.5 Kent State University

1.0.5.1 Description (Provided from the assignment instructions)

The producer generates items and puts items onto the table. The consumer will pick up items. The table can only hold two items at the same time. When the table is complete, the producer will wait. When there are no items, the consumer will wait. We use semaphores to synchronize producer and consumer. Mutual exclusion should be considered. We use threads in the producer program and consumer program. Shared memory is used for the “table”.

1.0.6 Implementation

Both the producer and consumer have their own processes, with a thread in each created and pointed to a function of the same name (i.e. consumer and producer respectively). With each thread running their own functions, a shared memory buffer with a capactiy to hold two items is shared between the producer and consumer. To help prevent deadlock and other data hazards when it comes to a shared buffer, a semaphore is used to indicate use of the bufffer. The number '1' indicates the buffer is currently in use, while the number '0' indicates the buffer is available for use.s

1.0.7 Compilation

Both the producer and consumer are seprate files and meant to be seprate processes. There are three options within the make file. The first two being consumer and producer respectively. These compiler the producer and consumer. The last option is clean, which cleans any object (.o) files as well as the compiled program files to be executed.

Running the consumer and/or producer seprately will not be of benifit as each relies on each other to run concurrently.

1.0.7.1 Consumer:

The following make command will compile the consumer files for execution

```
make consumer
```

1.0.8 Producer:

The following make command will compile the producer files for execution

```
make producer
```

1.0.9 Clean:

The following make command will remove the compiled and executable program files

```
make clean
```

1.0.10 Run Instructions:

To run both of the producer and consumer at the same time, for Linux and Unix systems, the following command is used

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

1.0.11 Libraries & Tech Stack

- C++
- Make
- G++
- PThread Library
- SharedMem 3 Library