Ian Altoveros

CS433 Operating Systems

Professor Zhang

November 30th , 2020

Programming Assignment 4

In this programming assignment, I tackled the programming solution to a bounder-buffer problem using a producer and consumer process shown in the textbook. In my program, I used three semaphores which are

sem\_t empty;

sem\_t full;

pthread\_mutex\_t mutex;

These three semaphores help the program in which it protects whether or not an item could be insert or deleted in the buffer. For example, if an item is being inserted into the buffer and it’s full it will not allow it to be inserted. Same with the mutex lock, the program will check if it’s locked or unlocked before inserting an item into the buffer. Next, If the program actually produces an item and is successfully able to put it into the buffer array it will print out on the screen the item that it produced which is randomly generated by the RAND\_MAX variable. Then if we look at the consume function of the program it will consume an item from the buffer array and display the newly outputted buffer. For inserting an item, I inserted the item into the first slot of the array and changed where the tail is located by adding + 1 to the current tail while taking the % BUFFER\_SIZE of it then increasing the count of the array. For deleting an item, I changed the location of the head by doing this (head % 1) % BUFFER\_SIZE; which allows me to change the location of the current head of the array then I proceeded to minus the current count of the array.