

Program 2

I. Problem Statement

The assigned task was to design, implement, and test a program in C/C++ that solves the Producer/Consumer problem by translating figure 5.14 into pthreads implementation and semaphores. It was also required to use five consumers instead of one. Producers were to supply the next long long value, originally initialized to zero. The main was to terminate the threads after 5 minutes.

II. Approach

The decision was made to develop the program in C due to programmer familiarity and its relatively low resource consumption compared to C++. The program was developed in Visual Studio Code on Ubuntu 20.04 LTS. The program was made so that the main process creates the consumer threads, then the producer thread. Then it waits for five minutes and terminates the threads. The producer produces values and sends them to the consumer, and the consumer consumes them. To find out how many times a given consumer consumes, the decision was made to have it output a Microsoft Excel file with the count of the times each consumer consumed.

III. Solution

The assigned task was to design, implement, and test a program in C/C++ that uses pthreads and semaphores to solve the Producer/Consumer problem. First, the main function gets the start time of the program and initializes the semaphores. Then it creates the consumer threads in an array and the producer thread on its own. Then the main waits in an infinite loop, continually comparing the time elapsed with the total time (5 minutes).

The producer initializes the long long to zero. It then enters an infinite loop of producing the next long long, sleeping for 15000 microseconds (this is done to throttle the program, to ensure it does not use too much space), and sending the product. The consumer has a loop of taking the data, consuming the data, and increasing the count of times it has consumed. Once five minutes have elapsed, the main function outputs the consume counts into a Microsoft Excel file then kills all the threads. Lastly, it destroys the semaphores and ends the program.

We can build, run, and test the program in Linux's terminal. First, install gcc if it is not already installed. Second, navigate to the directory where the file is stored in. Third, compile the program using the command 'gcc -o pgm2 -pthread bsa190001_program2.c'. Fourth, run the program in the terminal. This can be done with the command './pgm2'. Once that is done, the

program should execute on its own. Once the program has finished executing, it will have created a file called output.xlsx which contains the number of times each consumer has consumed. Figure 1 demonstrates how the program was compiled and tested, as well as the contents of output.xlsx after the test run. The output of the test shows that all required functionality works properly.

```

brandon@brandon-VirtualBox:~/Documents/OS_Code/Pgm2$ gcc -o pgm2 -pthread bsa190001_program2.c
bsa190001_program2.c: In function 'consumer':
bsa190001_program2.c:81:15: warning: cast from pointer to integer of different size [-Wpointer-to-int-cast]
    81 |         times[(int)arg]++; //Increase the count of times this consumer has consumed
        |         ^
bsa190001_program2.c: In function 'main':
bsa190001_program2.c:104:46: warning: cast to pointer from integer of different size [-Wint-to-pointer-cast]
    104 |     pthread_create(cth + i, NULL, consumer, (void*)(i+1));
        |                                     ^
brandon@brandon-VirtualBox:~/Documents/OS_Code/Pgm2$ ./pgm2
Consumer thread 1 has started
Consumer thread 2 has started
Consumer thread 3 has started
Consumer thread 4 has started
Consumer thread 5 has started
Producer thread has started
Consumer thread 1 has ended
Consumer thread 2 has ended
Consumer thread 3 has ended
Consumer thread 4 has ended
Consumer thread 5 has ended
Producer thread has ended
Program is complete. Counts outputted to output.xlsx
brandon@brandon-VirtualBox:~/Documents/OS_Code/Pgm2$

```

Consumer ID	Times Consumed
1	3881
2	3881
3	3881
4	3881
5	3880

Figure 1. This image shows an example of the program being compiled, run, and tested. All functionality works as outlined by the program. The program is built using gcc, and the program shows its progress along the way. On the right side of the image, the contents of output.xlsx after the run are shown.

Figure 2 is a chart that depicts how many times each consumer consumed a value from the buffer. The values were very even, to the point that they are almost identical. This is likely due to how the program tries to allow each consumer to retrieve it, as well as the extreme throttling. If the throttling were to be decreased, a larger variation would be seen between different consumers.

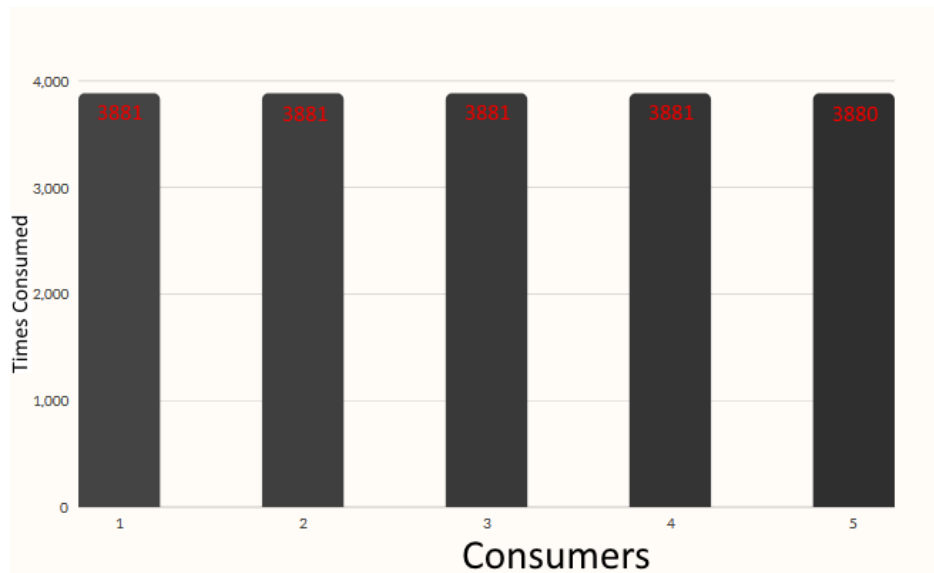


Figure 2. This image shows a bar chart depicting how many times each consumer consumed a value from the buffer. Across the horizontal axis is the consumers, numbered 1 through 5. The vertical axis is labeled "Times Consumed", and represents the number of times each consumer has consumed a value. At the top of each bar is a number denoting the exact value of the bar. Consumers 1 through 4 have a value of 3881, and consumer 5 has a value of 3880.