Christian Everett

[ceverett@cs.uml.edu](mailto:ceverett@cs.uml.edu)

Write-Up

In assignment two my degree of success was 100%. This assignment was compiled and run on mercury.cs.uml.edu. My approach to the problem was an event counter and sequencer solution. This program was similar to assignment 1 (another consumer-producer problem) but instead of using different processes for the producers and consumers there was only one process and multiple threads. I was able to take the example code file and write the code for the producer and consumer functions. 30 producers and 50 consumers were used in this program and each consumer ran until it had collected 200 dozen donuts (the producers ran forever). The output of each threads dozens of donuts are in files c1 for thread 1 up to cn for thread n.

The first requirement was to find the average running time for both process scope and system scope. The average runtime for process scope came to about 11.422 seconds, and system scope came to 17.42 seconds. This demonstrated forcing all threads to the same core improved performance. Then we were required to find the 50% deadlock queue size as in assignment 1. This came to be 57 as seen in table 1 (and in graph 1). With the 50% deadlock queue size I created another graph keeping the queue size at 575 and changing how many dozens each consumer was supposed to collect (graph and table 2).

Table 1

|  |  |
| --- | --- |
| Queue Size | Deadlock % |
| 400 | 86% |
| 500 | 66% |
| 575 | 50% |
| 600 | 33% |
| 700 | 20% |

Graph 1

Table 2

|  |  |
| --- | --- |
| Number Consumers | % Deadlocked |
| 100 | 6% |
| 150 | 40% |
| 250 | 66% |
| 300 | 69% |

Graph 2