HARIKRISHNAN GOPAL JANAKIRAMAN

PROJECT -2 (MULTIPLE PRODUCER AND CONSUMER)

[HARIKRISHNAN\_GOPALJANAKIRAMAN@STUDENT.UML.EDU](mailto:HARIKRISHNAN_GOPALJANAKIRAMAN@STUDENT.UML.EDU)

This Project is based upon creating multiple producers and multiple consumers and their relationship with deadlock. The producer will be producing four different types of donuts and are filled in four different types of ring buffers with the help of random number generator. The consumer consumes these dozen donuts with the help of random number generator. Because of multiple producers and multiple consumers, thread concepts are used to function effectively.

The program used to run effectively around 82%. Efficiency of the program is mainly dropped due to the less system space availability in the college server due to usage by many students.

The other problems are like: In certain areas, I used ineffective coding’s, so I referred some websites and get help from my friends to overcome those issues.

This program follows this typical rule:

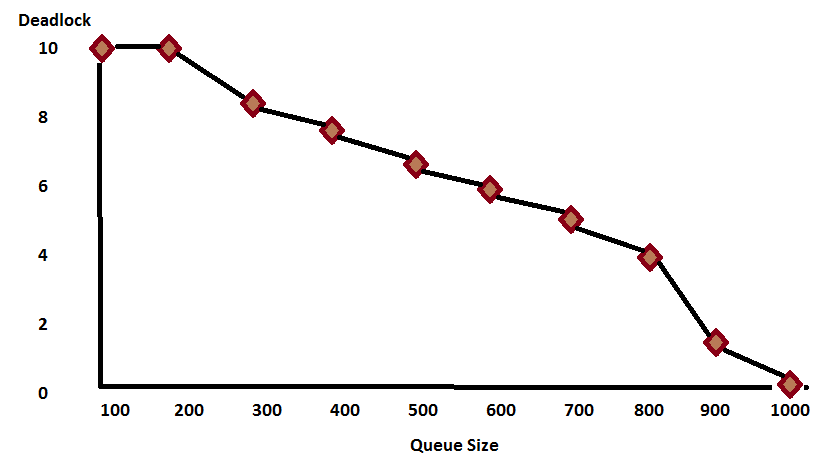
1-> No more donuts production from the producers if the ring buffer is full

2->No more donut consumption by the consumer if the ring buffer is empty.

The program output is analyzed with the help of two graphs by keeping 50 producers and 30 consumers as constant and they are plotted as follows:

Graph-1:

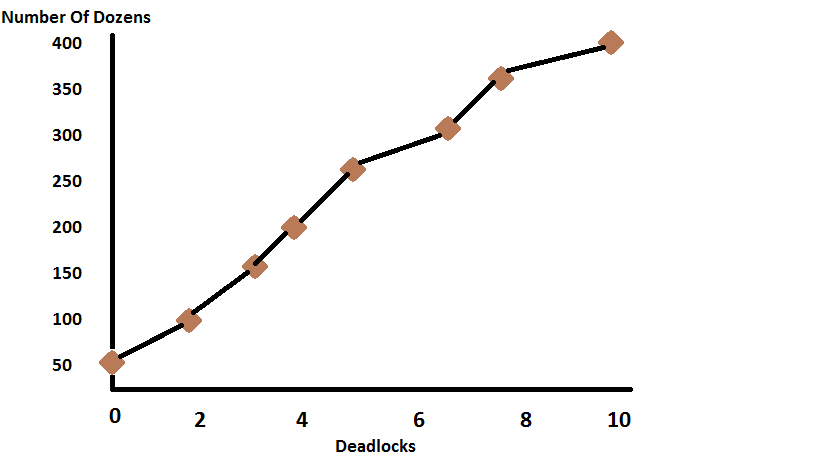
Analysis of Deadlock by Varying Queue size.



By Keeping Number of consumers and producers constant, the deadlock started to decrease with the increase in the queue size. That is, 50% deadlock occurred when queue size is 600 and it is considered in the next graph.

Graph-2:

Varying number of dozens Vs. deadlocks by keeping queue Size – 600 as constant.



Based upon this graph, it is clear that number of deadlocks increase with the number of dozens by keeping queue size as constant to 600.