Christian Everett

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

Write-Up

In assignment three my degree of success was 100%. My implementation used Ricarts algorithm and was split into two different processes a Node processes on each host; and a buffer manger process on one host. Each node process had one node controller thread and some number of producer and consumer threads. The node controller thread would monitor some number of file descriptors using the linux epoll. Output is provided for 5 of the consumers in the 5 producer 15 consumer scenario. Using that same configuration, I graphed the queue size to deadlock % (see table 1). This table will show I observed a queue size of 70 with a 50% deadlock. With this queue size I then created another graph with the number of total consumer threads vs the deadlock % (table 2). As the number of consumer threads increased deadlock percentage also increased.

Table

|  |  |
| --- | --- |
| Queue Size | Deadlock % |
| 100 | 10 |
| 90 | 30 |
| 80 | 40 |
| 70 | 50 |
| 50 | 100 |

Graph

Table

|  |  |
| --- | --- |
| Number Consumers | % Deadlocked |
| 3 | 10 |
| 10 | 20 |
| 20 | 70 |
| 30 | 80 |

Graph