Assignment 2

Experiment 1:

No. of loops: 10

No. of Producer: 30

No. of Consumer: 50

No. of slots: 600

No. of dozens: 50

|  |  |
| --- | --- |
| Process Scope | System Scope |
| 0.338242 | 0.306367 |
| 0.241018 | 0.267946 |
| 0.273579 | 0.245820 |
| 0.263716 | 0.274125 |
| 0.271470 | 0.243535 |
| 0.267800 | 0.252322 |
| 0.233731 | 0.231284 |
| 0.209211 | 0.207851 |
| 0.213402 | 0.225244 |
| 0.228321 | 0.215933 |
| Average: 0.232709 | Average: 0.247043 |

As you see, in the above experiment, the average of 10 loops for process scope is 0.232709 seconds, while the average of 10 loops for system scope is 0.247043 seconds. From the result we can conclude that the program executes faster on multiple cores (system scope) than on single core (process scope). This is because, in multiple core the threads can begin its execution independent of another thread, if it receives the lock and does not have to wait, which is not the same in single core.

Experiment 2:

No. of loops: 10

No. of Consumers: 50

No. of Producers: 30

No. of dozens: 200

|  |  |
| --- | --- |
| No. of slots (Queue size) | Deadlock % |
| 300 | 90 |
| 400 | 80 |
| 450 | 70 |
| 500 | 70 |
| 550 | 70 |
| 600 | 50 |
| 650 | 30 |
| 700 | 10 |
| 750 | 10 |
| 800 | 0 |

X-axis: No. of slots

Y-axis: Deadlock%

In the above experiment, we have kept the no. of producer, no, of consumer, no. of loops and no. of dozens constant. We find 50% deadlock when the queue size is 600, i.e. when the number of slots is 600. On illustrating the graph, we find out, that with increase in the number of queue size, there is a decrease in the deadlock %.

Experiment 3:

No. of loops: 10

No. of Consumers: 50

No. of Producers: 30

No. of slots: 600

|  |  |
| --- | --- |
| No. of dozens | Deadlock % |
| 50 | 0 |
| 100 | 10 |
| 150 | 40 |
| 200 | 50 |
| 250 | 70 |
| 300 | 80 |
| 350 | 80 |

X-axis: No. of dozens

Y-axis: Deadlock%

We observe that on reducing the number of dozens, the chances of deadlock occurring is low. But as the number of dozen increases, the deadlock rate also increases. The deadlock was found to be 50% when the number of dozens is 200. In the above experiment, we have kept no. of loops, no. of producers, no. of consumers and no. of slots constant. So we conclude, as the dozen size increases, the chances of getting a deadlock also increases i.e. the deadlock % also increases.

Success Rate: The success rate of the program is around 97%

Approach:

In this assignment, the first experiment that we do is to compare the Process scope and System scope.

Process scope is basically executing the program in a single core, while system core is when the program is running in multiple cores. We find out that program runs faster in multiple cores. We run the program on single core which has 0 deadlocks, and note down the timing for all the 10 loops. We then replace the call "PTHREAD\_SCOPE\_SYSTEM" with "PTHREAD\_SCOPE\_PROCESS" to find the same, with multiple cores in ‘pc\_threads.c’.

The second and third experiment is comparing the deadlock rate varying no. of slots and no. of dozens respectively. We find the 50% deadlock when no. of slots is 600 and when no. of dozens is 200 respectively.

I rated the success as 97% because I couldn’t find much of a difference between the system scope and process scope.

E-mail ID: Hetali\_ShahGala@student.uml.edu