Bilkent University

Department of Computer Engineering

Project #3 Report

Synchronization and Deadlocks

CS 342 - Operating Systems - Spring 2019

Section 3

April 21, 2019

Bartu Atabek	21602229
Hygerta Imeri	21603212

Experiments - comments and observations:

The following experiments were done with the stated values over 3 different mechanisms; DEADLOCK_NOTIHING, DEADLOCK_AVOIDANCE and DEADLOCK_DETECTION in order to observe the overhead and cost of these algorithms.

<u>Experiment 1:</u>

Successfully completed for DEADLOCK_AVOIDANCE and DEADLOCK_NOTHING

Parametric values and experiment results are as below:

N = 4

M = 3

exist[3] = {10,5,7};

max_demand = {5,5,5}

req1 = {2,3,2}

req2 = {3,2,3}

request 1

release 1

request 2

release 2

NOTHING: 9.81 second

AVOIDANCE: 10.56 second

DETECTION: 9.98 seconds

<u>Experiment 2:</u>

Successfully completed for DEADLOCK_AVOIDANCE and DEADLOCK_NOTHING

Parametric values and experiment results are as below:

N = 4

M = 5

exist[5] = {10,10,10,10,10};

max_demand = {2,2,2,2,2}

req1 = {0,0,0,0,0}

req2 = {1,1,1,1,1}

request

request

release

release

NOTHING: 17.69 second

AVOIDANCE: 35.43 second

DETECTION: 33.46 seconds

Experiment 3:

Successfully completed for DEADLOCK_AVOIDANCE and DEADLOCK_NOTHING

Parametric values and experiment results are as below:

N = 5

M = 3

exist[3] = {25,25,25};

max_demand = {20,20,20}

req1 = {10,10,10}

req2 = {5,5,5}

req2 = {1,1,1}

request 1

request 2

release 1

request 3

release 2

release 3

NOTHING: 12.21 second

AVOIDANCE: 24.28 second

DETECTION: 23.59 seconds

Experiment 4:

Successfully completed for DEADLOCK_AVOIDANCE and DEADLOCK_NOTHING

Parametric values and experiment results are as below:

N = 2

M = 3

exist[3] = {5,5,5};

max_demand = {3,2,0}

req1 = {0, 1,0}

req2 = {1,0,0}

request 1

release 1

request 2

release 2

NOTHING: 28.67 second

AVOIDANCE: 39.34 second

DETECTION:38.94 seconds

Experiment 5:

Successfully completed for DEADLOCK_AVOIDANCE and DEADLOCK_NOTHING

Parametric values and experiment results are as below:

N = 4

M = 6

exist[6] = {6,6,6,6,6,6};

max_demand = {5,5,5,5,5,5}

req1 = {5,5,5,5,5,5}

request 1

release 1

NOTHING: 1 min 4 seconds

AVOIDANCE: 1 min 22 seconds

DETECTION: 1 min 18 seconds

Experiment 6:

Successfully completed for DEADLOCK_AVOIDANCE and DEADLOCK_NOTHING

Parametric values and experiment results are as below:

N = 10

M = 6

exist[6] = {5,5,5,5,5,5};

max_demand = {3,2,1, 3,2,1}

req1 = {1,1,1,1,1,1}

req2 = {1,0,0,0,0,0}

request 1

release 1

request 2

release 2

NOTHING: 1 min 35 seconds

AVOIDANCE: 1 min 58 seconds

DETECTION: 1 min 53 seconds

Observations:

In a comparison between 'doing nothing' and 'deadlock avoidance' we looked at the time results to decide on the overhead of deadlock avoidance. The results show that indeed avoiding a deadlock causes more overhead than doing nothing against it. That is because of the continuous safety checks, which include a bunch of if statements and for/while loops that need to be taken into consideration while thinking of the possible occurrence of a deadlock in the future.

In addition, if we were to compare the overhead cost of deadlock avoidance and deadlock detection, then the results are similar with slight differences, meaning the overhead cost of deadlock detection is slightly smaller than that of deadlock avoidance. These results are normal when we think of the general logic of the algorithms, as a part of deadlock avoidance algorithm resembles deadlock detection in the way it checks whether the processes will face a deadlock or not. Furthermore, if it does, then it needs to go over some other checks which result in further overhead.