Matt Ritchie

CS420

HW5

1) Consider the deadlock situation that could occur in the dining-philosophers problem when the philosophers obtain the chopsticks one at a time. Discuss how the four necessary conditions for deadlock indeed hold in this setting. Discuss how deadlocks could be avoided by eliminating any one of the four conditions.

A chopstick can only be used by one philosopher at a time, which is mutual exclusion. Hold and wait is…each philosopher holding one chopstick and waiting to pick up another. Preemption doesn’t exist because no philosopher can demand a chopstick from another in this case. And if every philosopher at the table are waiting on a chopstick from the guy to their left then nobody moves and that’s circular wait.

Without mutual exclusion, one chopstick could be shared between all philosophers and they could all eat. Without hold and wait, nobody would touch any chopstick until they could grab both at once, which would never happen. No progress, but not technically a deadlock either. If they could demand each other to put down a stick, that would enable preemption which would break all the deadlocks.

2) Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock-free.

No cycles, no deadlocks.

3) Consider the following snapshot of a system:

Process Allocation Max Available

------- ------------ ----------- -----------

A B C D A B C D A B C D

------- ------------ ----------- -----------

P0 0 0 1 2 0 0 1 2 1 5 2 0

P1 1 0 0 0 1 7 5 0

P2 1 3 5 4 2 3 5 6

P3 0 6 3 2 0 6 5 2

P4 0 0 1 4 0 6 5 6

Answer the following questions using the banker’s algorithm:

a) What is the content of the matrix Need?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| P0 | 0 | 0 | 0 | 0 |
| P1 | 0 | 7 | 5 | 0 |
| P2 | 1 | 0 | 0 | 2 |
| P3 | 0 | 0 | 2 | 0 |
| P4 | 0 | 6 | 4 | 2 |

b) Is the system in a safe state? If so, provide a sequence that would allow all processes to run to completion

Execute Resources available after completion

-P0 first,

Run to completion <1, 5, 3, 2>

-P2->complete <2, 8, 8, 6>

-P4->complete <2, 8, 9, 10>

-P3->complete <2, 14, 12, 12>

-P1->complete <3, 14, 12, 12>

c) If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately (i.e. would the system still be in a safe state if the request were granted)? If so, provide a safe sequence that would allow all processes to run to completion.

Yup, no problem for the request.

Allocation for P1 would become <1, 4, 2, 0> so P1‘s Need calculates to <0, 3, 3, 0>. This would make available resources <1, 1, 0, 0>. Sequence:

Execute Resources available after completion

-P0 first,

Run to completion <1, 1, 1, 2>

-P2->complete <2, 4, 6, 6>

-P1->complete <3, 8, 8, 6>

-P4->complete <3, 8, 9, 10>

-P3->complete <3, 14, 12, 12>