University of Calgary

CPSC 457: Principles of Operating System, Winter 2018

Assignment 4

For
Coskun Sahin, Dr.Pavol Federl

By
BenKun Chen
30005337
benkun.chen@ucalgary.ca
Tutorial Section: T02

Q1 – Written question (5 marks)

Consider 3 processes A, B and C which want to perform operations on resource R with 1 instance:

The table below shows the case when the process will lead to a deadlock. Since process ‘A’ is the only process that can execute, the new ‘Available’ is 1. However, the process can’t proceed since 1 is smaller than both process ‘A’ and ‘B’s ‘Need’. So that, sequence ‘A’ leads to a deadlock.

Process	Allocation	Max	Available	Need
A	1	1	0	0
B	0	2		2
C	0	2		2

If we lower the ‘Max’ for resource in processes ‘B’ and ‘C’ by 1, then we only need 1 resource of ‘Need’ for processes ‘B’ and ‘C’. Again, the process ‘A’ will execute first and the new ‘Available’ will be 1. Since the ‘Need’ for both process ‘B’ and ‘C’ satisfy the new ‘Available’. Then, sequence A -> B -> C could be executed successfully. So that, it leads to the completion of all processes.

Process	Allocation	Max	Available	Need
A	1	1	0	0
B	0	1		1
C	0	1		1

Q2 – Written question (5 marks)

First, we calculate the 'Need' for each process:

Process	Allocation	Maximum	Available	Need (Max - Alloc)
P0	1 0 2 1 1	1 1 2 1 3	0 0 x 1 2	0 1 0 0 2
P1	2 0 1 1 0	2 2 2 1 0		0 2 1 0 0
P2	1 1 0 1 0	2 1 3 1 0		1 0 3 0 0
P3	1 1 1 1 0	1 1 2 2 1		0 0 1 1 1

Now we try to make x = 0:

Process	Allocation	Maximum	Available	Need (Max - Alloc)
P0	1 0 2 1 1	1 1 2 1 3	0 0 0 1 2	0 1 0 0 2
P1	2 0 1 1 0	2 2 2 1 0		0 2 1 0 0
P2	1 1 0 1 0	2 1 3 1 0		1 0 3 0 0
P3	1 1 1 1 0	1 1 2 2 1		0 0 1 1 1

We then start comparing the 'Available' with the 'Need'. We found that first 2 given digits from 'Available' can only satisfy the 'Need' for P3. Also, the third digit in P3 is bigger than the the corresponding digit in 'Available'. As a result, x = 1 can't keep the system in a safe state. Now we try to make x = 1:

Process	Allocation	Maximum	Available	Need (Max - Alloc)
P0	1 0 2 1 1	1 1 2 1 3	0 0 1 1 2	0 1 0 0 2
P1	2 0 1 1 0	2 2 2 1 0		0 2 1 0 0
P2	1 1 0 1 0	2 1 3 1 0		1 0 3 0 0
P3	1 1 1 1 0	1 1 2 2 1		0 0 1 1 1

When x = 1, x satisfies the requirement since it is equal to the third digit in 'Need'. Fourth and fifth digit also satisfy the requirement since 1 = 1 and 2 > 1.

One of the correct safety algorithm and sequence is:

1. P3: available = (0, 0, 1, 1, 2) + (1, 1, 1, 1, 0) = (1, 1, 2, 2, 2)
2. P0: available = (1, 1, 2, 2, 2) + (1, 0, 2, 1, 1) = (2, 1, 4, 3, 3)
3. P1: available = (2, 1, 4, 3, 3) + (2, 0, 1, 1, 0) = (4, 1, 5, 4, 3)
4. P2: available = (4, 1, 5, 4, 3) + (1, 1, 0, 1, 0) = (5, 2, 5, 5, 3)

Q3 – Written question (5 marks)

Please see attached file - “banker.cpp”