Evan Owre, CSCE311 HW 2, 10/13/20

CSCE 311 – Operating Systems

Homework 2

Deadlock and Contiguous Memory

40 Points

Assigned on: September 29th, 2020
Due: October 13th, 2020 @ 11:59 pm
Deliverable: PDF or Doc(x) or Odt of your answers to the following.

1. (20 points) Consider the following snapshot of a system:

	Allocation					Max					Available			
	A	**B**	**C**	**D**		**A**	**B**	**C**	**D**		**A**	**B**	**C**	**D**
P0	1	0	2	0		2	2	3	3		2	3	1	1
P1	1	0	1	2		3	5	3	5					
P2	0	1	0	2		2	2	1	3					
P3	3	2	1	0		6	3	3	5					
P4	0	1	0	1		3	3	1	3					

Answer the following questions using the Banker's Algorithm

a) Show the content of the Need matrix.

	Need			
	A	**B**	**C**	**D**
P0	1	2	2	3
P1	2	5	2	3
P2	2	1	1	1
P3	3	1	2	5
P4	3	2	1	2

b) Show that the system is in a safe state by listing the order in which processes can be executed without producing a deadlock.

Avalible 2 4 1 3 P2 done <P2,P0,P4,P1,P3>
3 4 3 3 P0 done
3 5 3 4 P4 done
4 5 4 6 P1 done

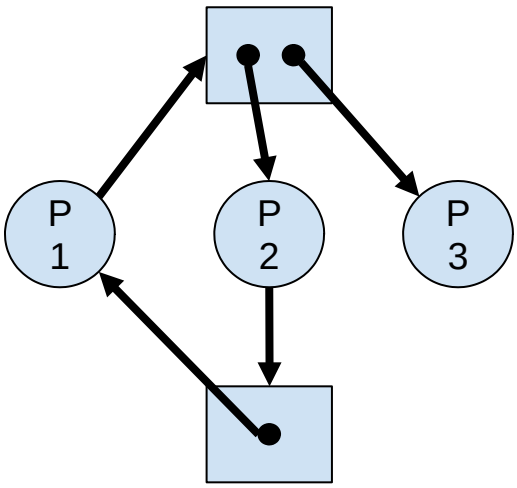
Evan Owre, CSCE311 HW 2, 10/13/20

7 7 5 6 P3 done

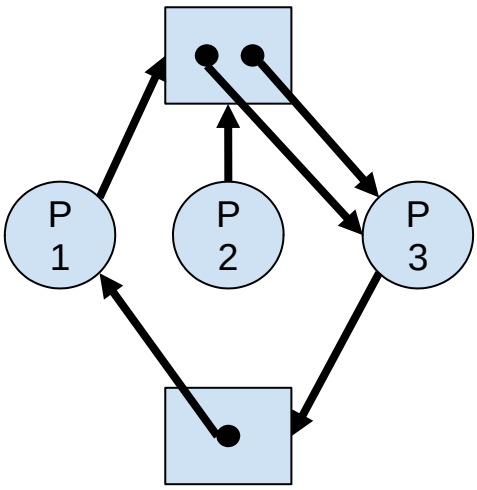
c) If P3 comes in with a request of (0,1,0,0), will its request be able to be processed immediately? If so, list a safe sequence for the processes to execute without producing deadlock. If not, list the processes that are possibly in a deadlock.

P3 0 1 0 0
Allocated 3 3 1 0
Need 3 0 2 5
Avalilible 2 2 1 1
No P3 can not be processed imdiedatly, proccesses P0, P1, P2, P4 will all be deadlocked

2. (5 points) Given the following resource allocation graphs, is there deadlock present? If so, list what processes are in deadlock. If not, list a safe sequence for the processes to execute.



A.



B.

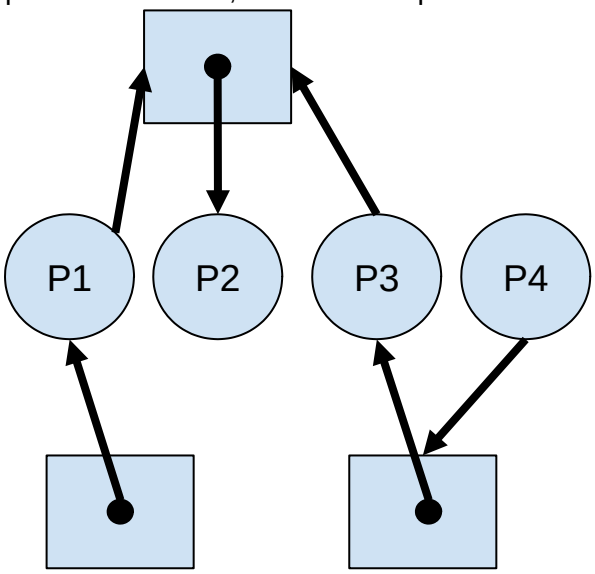
Answer for A:

Yes there is deadlock in the allocation graphs, P2 is waiting for the bottom resource which is allocated to P1, P1 is not in deadlock becasue it will get top resource from P 3

Answer for B:

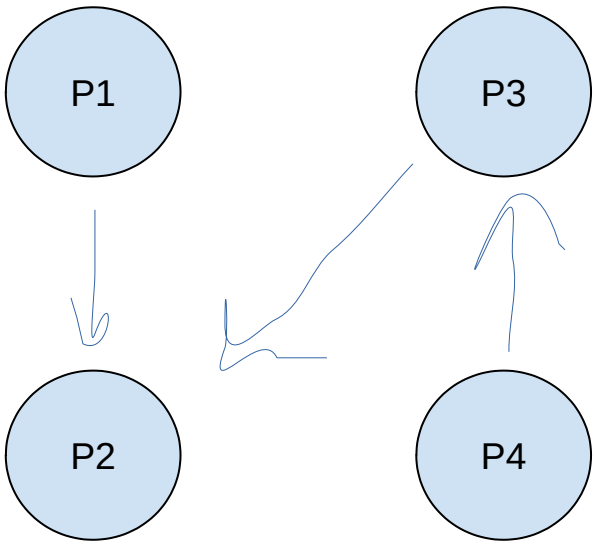
deadlock is present, P1 is waiting for top resource, allocated to P3
P2 is waiting for top resource, allocated to P3.
P3 is waiting for bottom resource, allocated to P3

3. (5 points) Convert the following resource allocation graph to a wait-for graph. Using the final wait-for graph, determine if there are any processes in deadlock. If so, which processes? If not, list a safe sequence of execution for the processes



Wait-for graph:
no cycles in the graph, no deadlock, safe sequence <P2, P1, P3, P4>

Evan Owre, CSCE311 HW 2, 10/13/20



4. (10 points) Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order - shown in figure below), how would the first-fit, best-fit, and worst-fit algorithms place processes of 75 KB, 499 KB, 50 KB, and 426 KB (in order)? If one process will not fit you can list their partition as NULL. Which algorithm makes the most efficient use of memory (use amount of internal fragmentation to decide)?

Partition	A	B	C	D	E
Size	100 KB	500 KB	200 KB	300 KB	600 KB

First-fit:

Partition	A	B	C	D	E
Process (list size)	50kb	75kb	Null	Null	426kb

Processes not scheduled in: 499kb

Best-fit:

Partition	A	B	C	D	E

Process (list size)	50kb	426kb	75kb	Null	499kb

Processes not scheduled in: ***none***

Worst-fit:

Partition	A	B	C	D	E
Process (list size)	Null	75kb	Null	Null	50kb

Processes not scheduled in: 426kb, 499kb