CS341 OS Quiz   Date 30th Oct 2014 Answer All Question

Q1[3 marks] : Write different page table organizations to handle big page table (without using virtual memory). No need to draw diagram.

Ans: (1) Hierarchical (or two level or multi level) page organization. Inner page and outer page, paging the page table
(2) Hashing
(3) Inverted Page Table, where number of frame << number of virtual page

Q2 [3] : Consider the following page reference string.   1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 3, 7, 7, 6, 3
How many page faults will occurs for FIFO, LRU and Optimal algorithm assuming 3 frames.

*Ans: FIFO : **1**, **2, 3**, **4**, 2, **1**, **5**, **6**, **2**, **1**, **3**, 7, 7, **6**, 3   : 12 page faults*
*LRU : **1**, **2, 3**, **4**, 2, **1**, **5**, **6**, **2**, **1**, **3**, 7, 7, **6**, 3   : 12 page faults*
*OPT: **1, 2, 3, 4,** 2, 1, **5, 6,** 2, 1, **3,** 7, 7, 6, 3 :   8 page fault*

Q3 [3] : What is stream data access? Calculate working set of Streaming data reference with (a) Δ=10 References and (b) Δ=20 References.

Ans: In stream data access, data are access sequentially without repetition.
Size of the working set with Δ=10 is 10 and Size of the working set with Δ=20 is 20.

Q4 [3]: What is the cause of thrashing? How does the system detect thrashing? Once it detect thrashing, what can the system do to eliminate this problem.

Ans: *A process is thrashing if it is spending more time paging then executing. Thrashing is caused by under allocation of the minimum number of memory frames required by a process, forcing it to continuously page fault. If D>m, it cause thrashing. Suppose a system, have m frames in memory. WSS_i is working set of process P_i and that is total number of pages referenced in the most recent Δ (varies in time). The total demand frames of all the running process is $D = \Sigma WSS_i$, (which is approximation of locality of all the running process in the system). If D > m then it implies there is a Thrashing problem.*

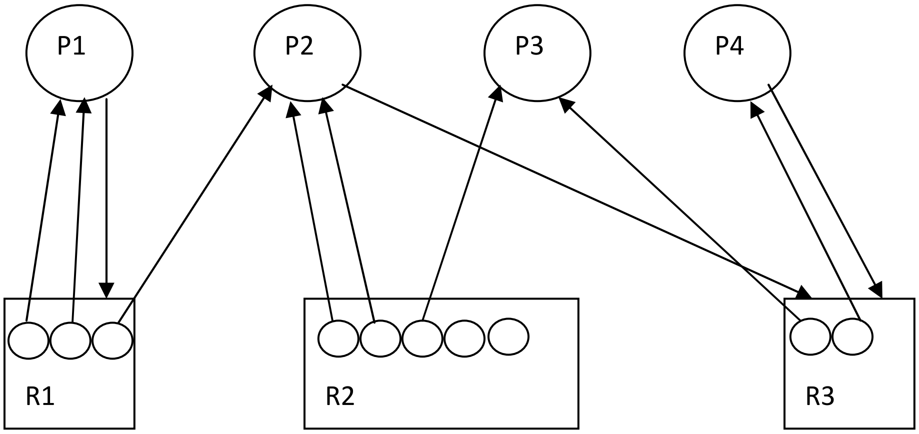
The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming. It can be eliminated by reducing the level of multiprogramming.

Q5 [2+2+1+1] : Draw Resource Allocation Graph for the system described in the following Table. Annotate edges (with X symbol) for reduction of the resource allocation graph.

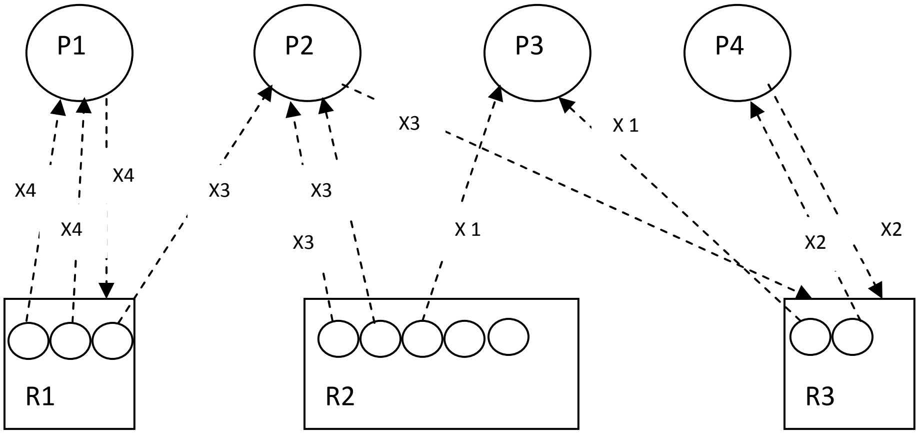
	Current Allocation			Outstanding Allocation			Maximum Allocation			Resource Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	2	0	0	1	0	0	2	0	1	0	2	0
P2	1	2	0	0	0	1	2	5	2			
P3	0	1	1	0	0	0	1	4	2			
P4	0	0	1	0	0	1	2	0	1			

Is the system deadlocked?
Is system in a safe or unsafe state?

Ans : Total number of instances of a resources = current allocation + available,   Total instances of R1=3, R2=5 and R3=2.



Reduced RAG with annotation is shown below: No process left to execute after reduction, so system in not in deadlock state. As maximum allocation is greater than current allocation plus the request or outstanding, so safety algorithm abort unconditionally will not work. Everyone will get 1 mark with condition that person have drawn the RAG.


P1
P2
P3
P4
X3
X 1
X4
X4
X3
X3
X4
X3
X 1
X2
X2
R1
R2
R3