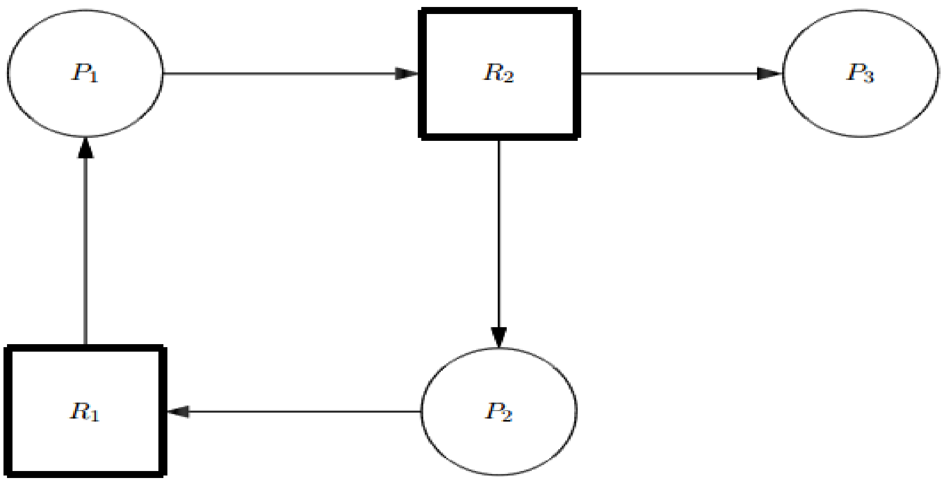
Amit Lawanghare Homework - 2

1. (10 points)
The following figure shows a resource graph for a system with consumable resources only. A resource is represented by a rectangle with thick lines and labeled as R_i. A process is represented by a circle, labeled P_i.
    (a) Is the graph a claim-limited graph? Why?
    (b) Is the graph reducible? Why?



Answer –

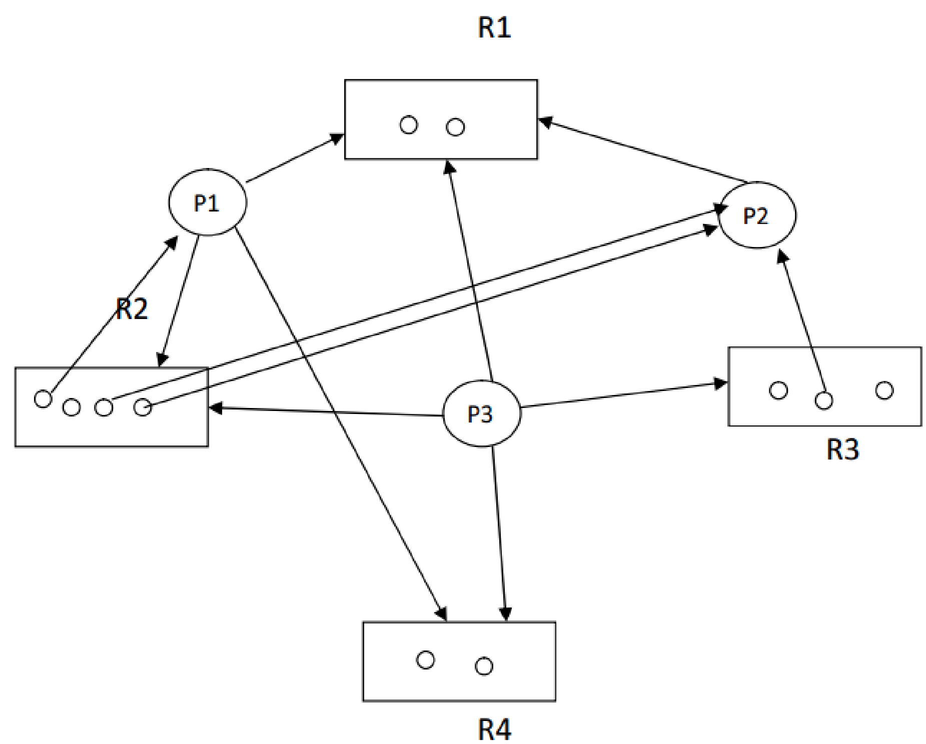
a. Since each resource has no more available unit and P1 is consumer of R1 whereas P2 and P3 are consumers of R2. This graph is claim-limited.

b. This graph is reducible because of following:
P1 is producer of R2 and since it is unblocked P1 can produce 2 units that P2 and P3 can consume. And P1 needs only one unit of R1 which can be produced by P2.
So all process requests can be granted hence this graph is reducible.

2. (10 points)
Assume a system has P processes and R identical units of a reusable resource. If each process can claim at most N units of the resource, determine whether each of the following is true or false and prove your claim:

(a) If the system is deadlock free then $R \geq P(N-1)+1$.
(b) If $R \geq P(N-1)+1$ then the system is deadlock free.

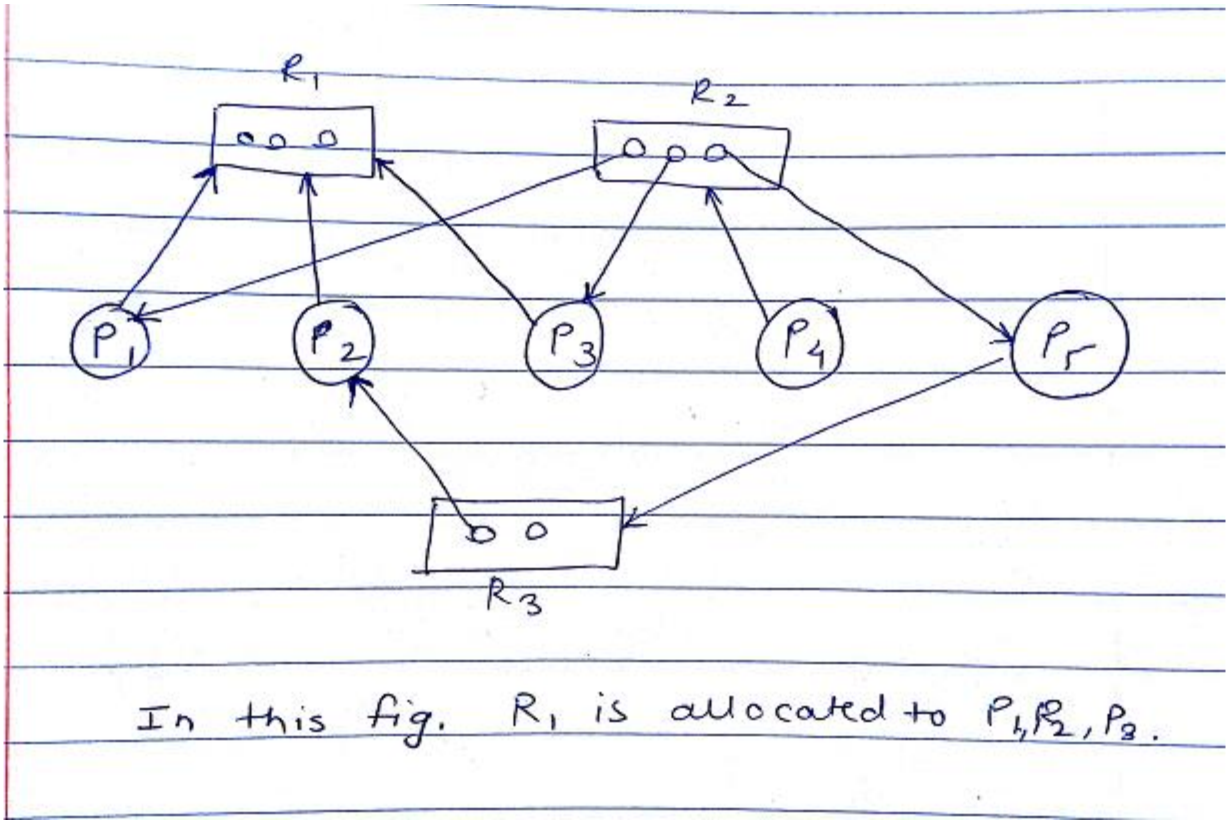
a. Let us assume that system is having R = P(N-1) as most. Now each process can hold N-1 units.



lets assume that there are 3 processes P_1, P_2, P_3.
and $N = 4$
So in above fig., total number of resources are 11
$P(N-1) = 3(4-1)+1 = 10$

so this graph is dedd lock free when $R \geq P(N-1)+1$

b.



As given in the diagram above.
R = 8
P = 5
N = 2

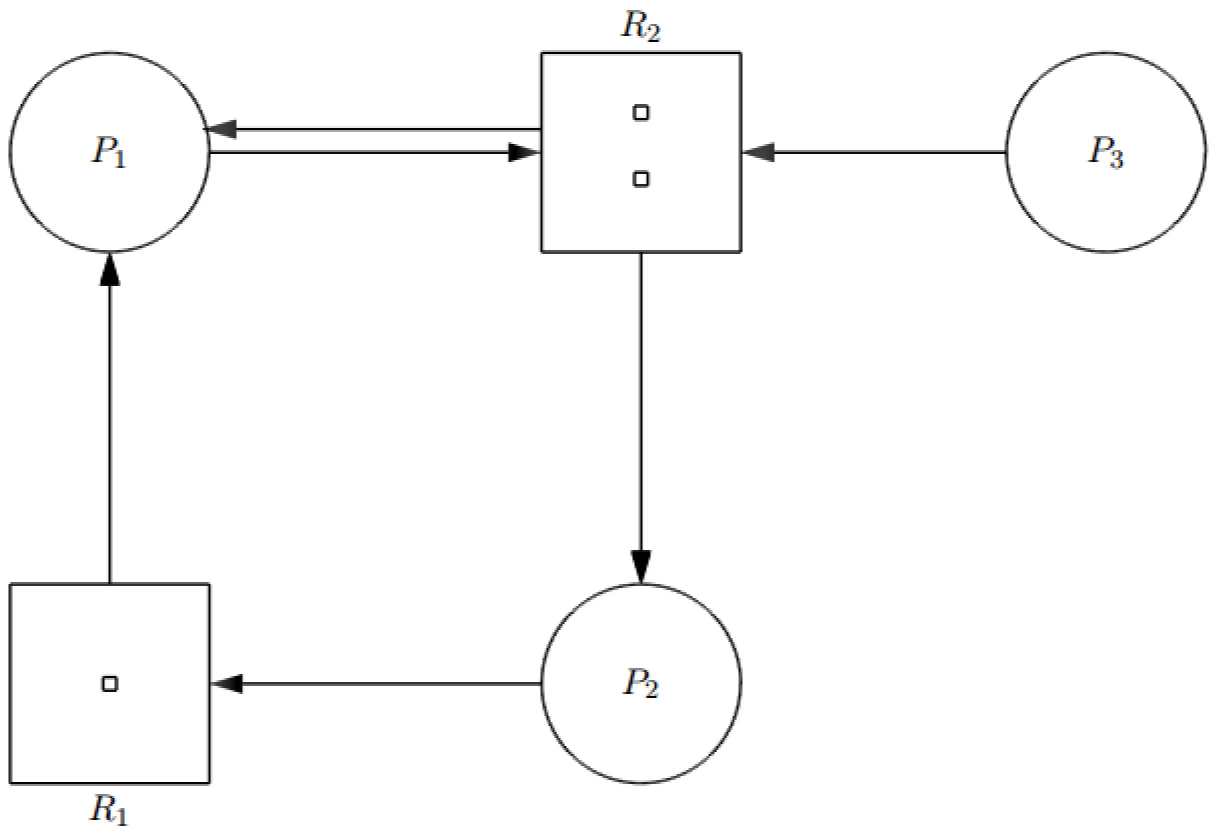
So here 8 >= 5(2-1) +1

Suppose, R1 is allocated to P1, P2 and P3. So, P1 can finish its work and release R2 and R2 can be allocated to P4. Now, P5 can also finish its work as the 2 resources are allocated so, P5 finishes its task and release R1 and R2. While coming to P2, R1 is already assigned and even R3 is assigned. So, it finishes its work. Similarly, P3 can finish its task by taking 1 R3 unit and 1 R2 unit.
So, when P3 completes execution P4 can continue.

3. (10 points)

The following figure shows a resource graph for a system with reusable resources only. A resource is represented by a rectangle, in which a small square indicates a unit of the resource.

(a) Is the graph expedient? Why?
(b) Is there any knot in the graph? Why?
(c) Is there any deadlock in the system? Why?



a. True. As all processes are having requests blocked which means graph is expedient.
b. True. All nodes in subgraph {p1,p2,p3,p4} are reachable from every other node this has knot.
c. True. As there is a knot present {p1,p2,p3,p4} it is sufficient for deadlock.

4. (10 points)

In this problem you are to compare reading a file using a single-threaded file server and a multithreaded server. It takes 15 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming that the data needed are in a cache in main memory. If a disk operation is needed, as is the case one-third of the time, an additional 75 msec is required, during which time the thread sleeps. How many requests/sec can the server handle if it is single threaded? If it is multithreaded?

In a single-threaded file server, it takes 15 msec to get a request for work
Does the rest of the necessary processing if the data is in a cache in main memory.
It takes additional one-third of the time, an additional 75 msec that is 90 msec if disk operation is needed.
Therefore 1/3(90) + 2/3(15) = 40 msec is the total time required for reading a file using single threaded file server.
So, the server can perform 25 requests/sec.
In multi-threaded file server, the waiting for disk is overlapped.

Therefore, it takes 15 msec for reading a file. So, in total it can perform 1000/15 requests per sec = 66(2/3) requests per second.

5. (10 points)
Consider the state of a system with processes P_1, P_2, and P_3, defined by the following matrices:

$$\text{max-Avail } A = \begin{pmatrix} 5 & 2 & 4 \end{pmatrix}$$
$$\text{max-Claim } B = \begin{pmatrix} 2 & 2 & 2 \\ 1 & 2 & 2 \\ 3 & 1 & 3 \end{pmatrix}$$
$$\text{Allocation } C = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

(a) Find the *available* matrix D and the *need* matrix E in this state.

(b) Suppose now process P_1 makes a request with

$$F_1 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$$

If the request were granted, what would be D, C, and E in the resulted state?

(c) To ensure the system be safe, should the request be granted? Why? Give your reasons in detail.

max avail $A = (5\ 2\ 4)$

" $B = \begin{bmatrix} 2 & 2 & 2 \\ 1 & 2 & 2 \\ 3 & 1 & 3 \end{bmatrix}$

$C = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

a) Available matrix $D = A - \sum_{k=1}^{n} C_k$

$D = (5\ 2\ 4) - (3\ 2\ 2)$

$D = (2\ 0\ 2)$

b $E = B - C$

$$\begin{pmatrix} 2 & 2 & 2 \\ 1 & 2 & 2 \\ 3 & 1 & 3 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 0 & 2 & 1 \\ 2 & 0 & 2 \end{pmatrix}$$

b) Suppose now P_1 makes request

$$F_1 = (0\ 0\ 1)$$

So

Available D = $D - F_1$

$$= (2\ 0\ 2) - (0\ 0\ 1)$$

$$D = (2\ 0\ 1)$$

Alloc^n $C = C_i + F_i$

$$= \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} + (0\ 0\ 1)$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$E = E_i - F_i$$

$$= \begin{pmatrix} 1 & 1 & 2 \\ 0 & 2 & 1 \\ 2 & 0 & 2 \end{pmatrix} - (0\ 0\ 1) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 2 & 0 & 2 \end{pmatrix}$$

The requst is granted when next state is safe state.

lets use safe-state check algorithm.

P_1 · $(1\ 1\ 1) \leq (2\ 0\ 1)$ false

P_2 $(0\ 2\ 1) \leq (2\ 0\ 1)$ false

P_3 $(2\ 0\ 2) \leq (2\ 0\ 1)$ false

so system is not in safe state as available matrix doesnt have enough resources given by need matrix.