Department of Computer Science and Engineering

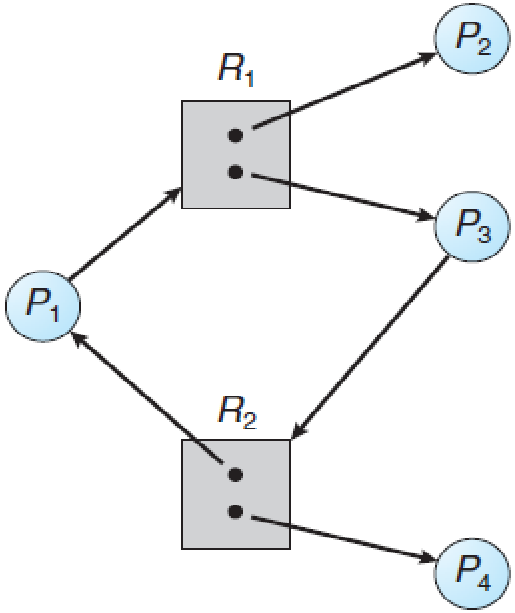
FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW
LUCKNOW

Dr. Zeeshan Ali Siddiqui
Assistant Professor
Deptt. of C.S.E.

RESOURCE-ALLOCATION GRAPH

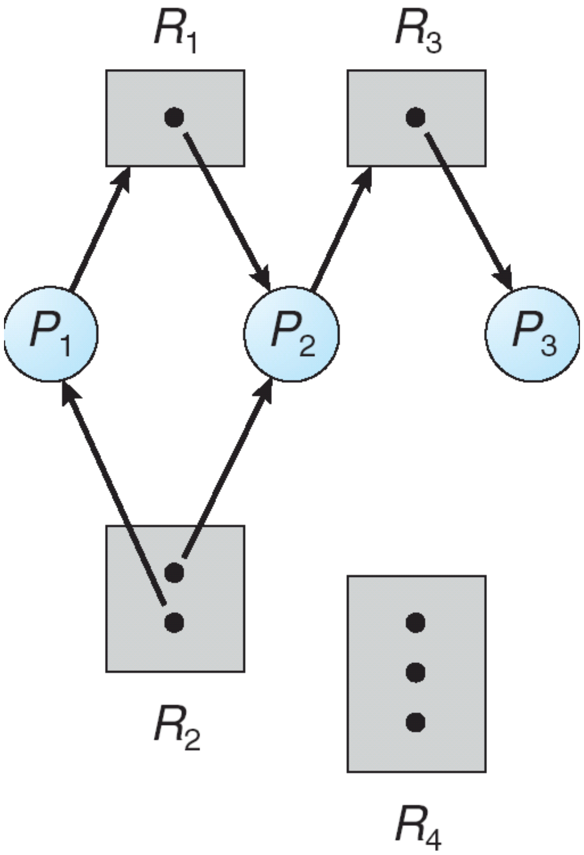
Resource-Allocation Graph 1/2

- This *graph* consists of a set of vertices V and a set of edges E.
- V is partitioned into two types:
  - P = {P1, P2, …, Pn}, the set of all the *processes* in the system
  - R = {R1, R2, …, Rm}, the set of all *resource* types in the system



- **Request edge** – directed edge Pi -> Rj (Pi has requested an instance of resource type Rj and is currently waiting for that resource).
- Assignment edge – directed edge Rj -> Pi (an instance of resource type Rj has been allocated to process Pi).

Example 1



Request edge

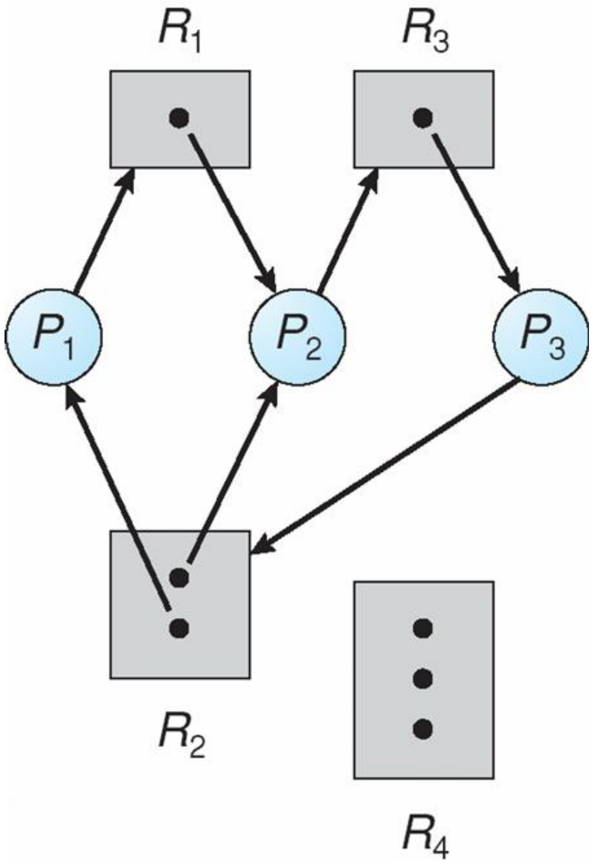
$P_1 \rightarrow R_1$, $P_2 \rightarrow R_3$

Assignment edge

$R_1 \rightarrow P_2$, $R_2 \rightarrow P_2$, $R_2 \rightarrow P_1$, $R_3 \rightarrow P_3$

No deadlock

Example 2



Request edge

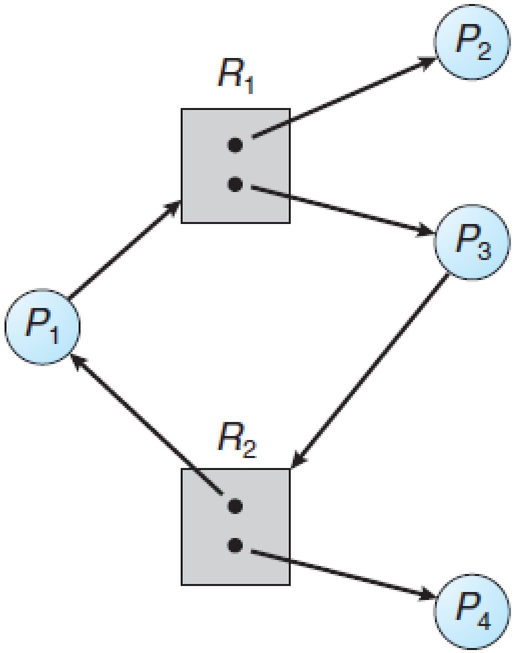
$P_1 \rightarrow R_1$ | $P_2 \rightarrow R_3$ | $P_3 \rightarrow R_2$

Assignment edge

$R_2 \rightarrow P_1$ | $R_2 \rightarrow P_2$ | $R_1 \rightarrow P_2$ | $R_3 \rightarrow P_3$

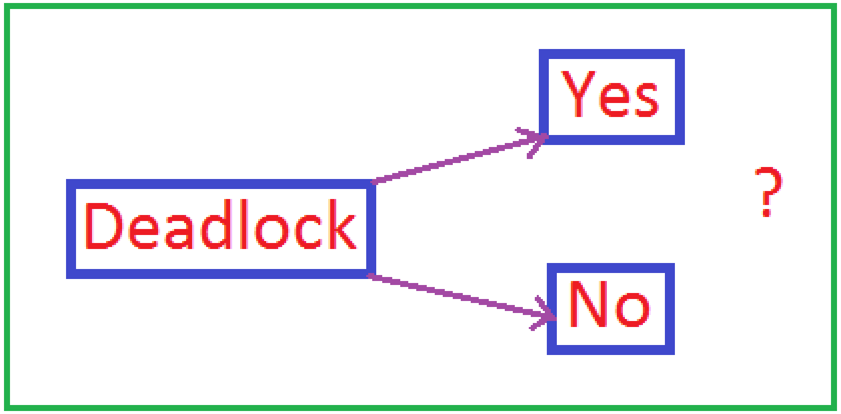
P1, P2, and P3: deadlocked

Question?



Request edge ?

Assignment edge ?



Deadlock

Yes

No

?

Basic Facts

- If graph contains *no cycles* -> **no deadlock**

- If graph contains a cycle

  - If only *one instance* per resource type, then *deadlock*

  - If *several instances* per resource type, *possibility of deadlock*.

References

1. Silberschatz, Galvin and Gagne, “Operating Systems Concepts”, Wiley.

2. William Stallings, “Operating Systems: Internals and Design Principles”, 6th Edition, Pearson Education.

3. D M Dhamdhere, “Operating Systems: A Concept based Approach”, 2nd Edition, TMH.

Thank You.