# Centralized DDD Algorithm

Assignment 06


1

## Ho- Ramamurthy DDD Algorithm


- Each site maintains resource status table and process status table.
  - Resource status table: Resources locked by processes.
  - Process status table: Processes that are locked or are waiting for resources.
- Controller periodically collects these tables from each site.

9 May 2025

2

## Ho- Ramamurthy DDD Algorithm

- Controller constructs a WFG from transactions common to both the tables.
- If there is no cycle, then no deadlock is detected.
- A cycle means a deadlock.



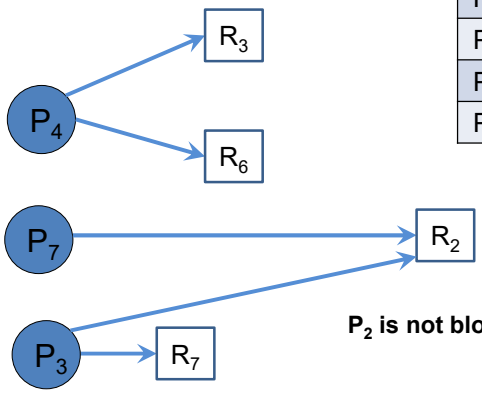


## Hints

- Start with 3 to 4 nodes in a graph
- Assume a set of resources and a set of processes for each node
- Populate two status tables for each of these nodes as described in the algorithm only for local resources and processes

# Process Status Table at Site 1

**Local Processes $P_2$, $P_3$, $P_4$, and $P_7$**




| | $R_2$ | $R_3$ | $R_6$ | $R_7$ |
|---|---|---|---|---|
| $P_2$ | 0 | 0 | 0 | 0 |
| $P_3$ | 1 | 0 | 0 | 1 |
| $P_4$ | 0 | 1 | 1 | 0 |
| $P_7$ | 1 | 0 | 0 | 0 |

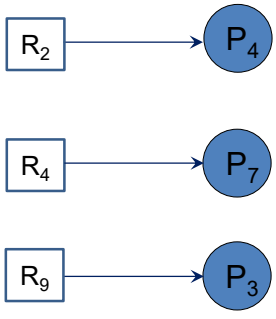
**$P_2$ is not blocked for any resource**

9 May 2025

5

# Resource Status Table at Site 1

**Local Resources $R_2$, $R_4$, and $R_9$**




| | $P_4$ | $P_7$ | $P_8$ |
|---|---|---|---|
| $R_2$ | 1 | 0 | 0 |
| $R_4$ | 0 | 1 | 0 |
| $R_9$ | 0 | 0 | 1 |

9 May 2025

6

## Hints


- Collect these status table data
- Build wait for graph matrix from process to resource request data and resource to process assignment data
- Check for cycle and infer accordingly