The University for world-class professionals

Manchester Metropolitan University

Deadlock Detection and Recovery

Emma Norling
John Dalton room E128
Email: E.Norling@mmu.ac.uk
Telephone: 0161 247 3884

Aims

By the end of this podcast, you should be able to:

- Explain the principle of deadlock detection and recovery
- Explain how to detect a deadlock
- Discuss deadlock detection strategies
- Discuss different recovery strategies

Deadlock Detection and Recovery

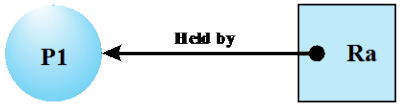
- Resource requests are granted whenever possible
- If deadlock arises, something is done about it

Checking for Deadlock

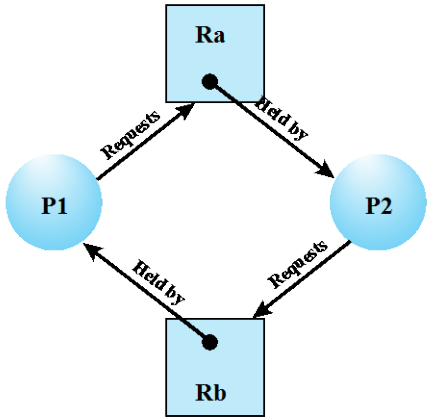
- A check for deadlock can be made as frequently as each resource request or, less frequently, depending on how likely it is for a deadlock to occur
  - the algorithm is relatively simple
  - frequent checks consume considerable processor time

Resource Allocation Graphs

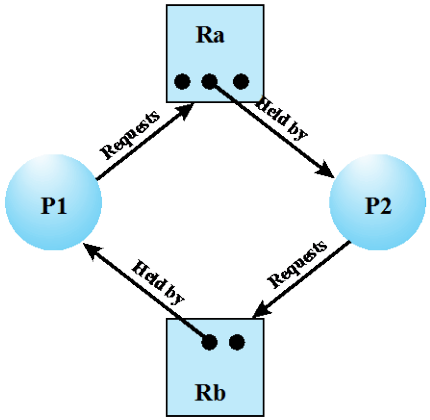
(a) Resouce is requested



(b) Resource is held



(c) Circular wait



(d) No deadlock

Example

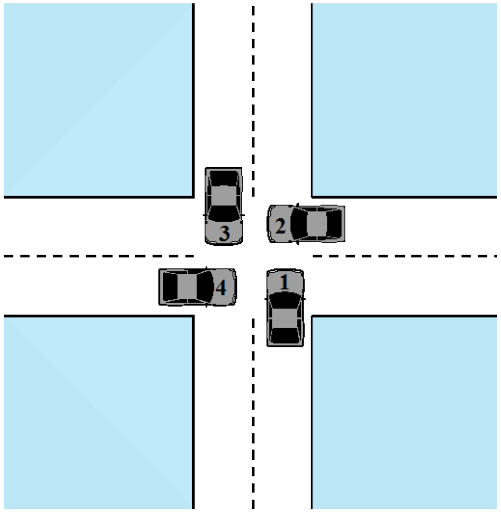


Figure 6.1 **(b) Deadlock**

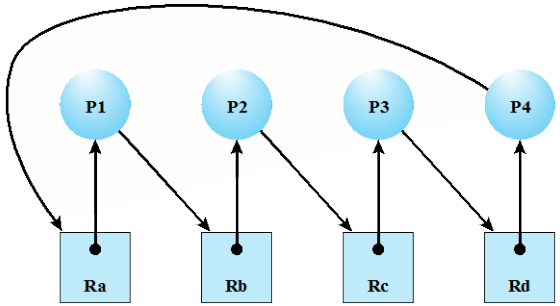
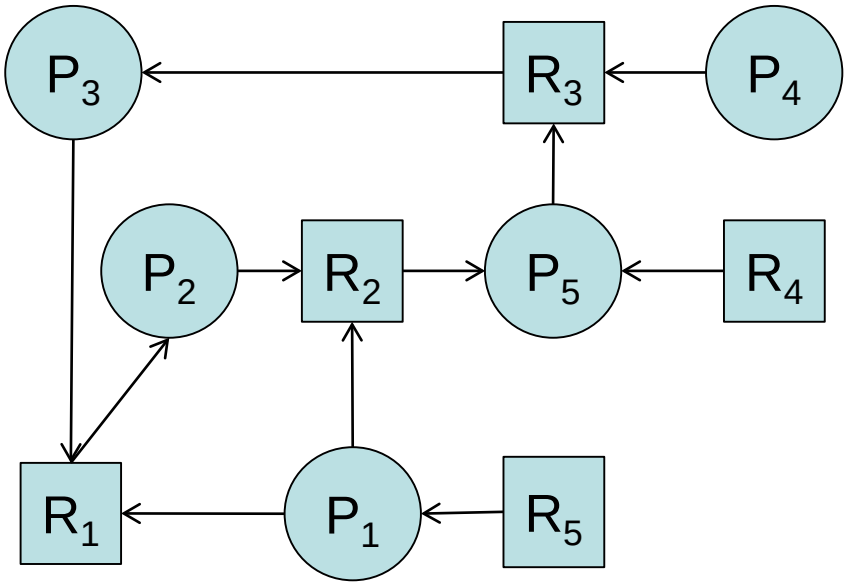
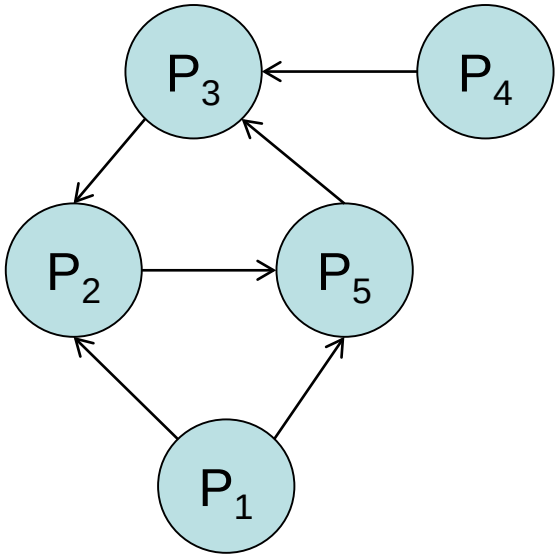


Figure 6.6 Resource Allocation Graph for Figure 6.1b

Wait-For Graph



Resource allocation graph



Wait-for graph

Using a Wait-For Graph

- When, and how often should it be invoked?
  - How often is a deadlock likely?
  - Infrequent invocation could obscure the original deadlock

Recovering from Deadlock

Three main alternatives:

- Process termination
- Resource pre-emption
- Backup to a checkpoint

Process Termination

- Abort all processes?
- Abort individual processes until deadlock eliminated?
  - In which order? Possibilities:
    1. Priority of the process
    2. How long process has computed, and how much longer to completion
    3. Resources the process has used
    4. Resources process needs to complete
    5. How many processes will need to be terminated
    6. Is process interactive or batch?

Resource Pre-emption

- Which process?
- How to rollback?
- Can cause starvation.

Backup

- Revert the entire system to a “safe” state
- Restart

Summary

- Deadlock detection and recovery makes good use of available resources
  - particular if deadlock is infrequent
- Detecting deadlock is simple but inefficient
  - RAGs/wait-for graphs
- Three alternative recovery strategies
  - Process termination
  - Resource pre-emption
  - Backup

What Next?

- Read the relevant section in the textbook (6.4)
- This concludes the podcasts for this week.