Review

- What is Deadlock?
- Resource Allocation Graph
  - Deadlock example with Resources Allocation Graph
- Resource Types for a Process
- Sequence for Resource Use
- Implementation of Resource request, use and release
- Deadlock Condition
- Four Strategies for Dealing Deadlock

Preview

- Deadlock detection and Recovery.
  - Detection with one resource of each type
  - Detection with multiple resource of each type
    - Detection Algorithm with multiple matrix
  - Recover from Deadlock after a deadlock detection

Four Strategies for Dealing Deadlock

- Just ignore
- Detection and Recover
- Dynamic Avoidance by careful allocation
- Prevention – by negating one of the four conditions necessary to cause deadlock

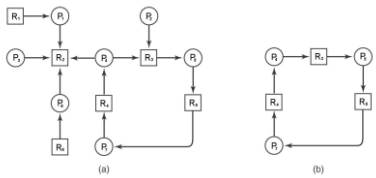
Deadlock Detection and Recovery

- When a process request a resource and if the resource is available, system let the process hold the resource.
- For each resource allocation, OS keeps tracking whether there is any deadlock or not (OS check a deadlock after each resource allocation).
- When a deadlock situation is detected by system, system takes some action to recover from the deadlock – deadlock detection and recovery from deadlock

Deadlock Detection and Recovery
(Detection with one resource of each type)

- Deadlock Detection with one resource of each type is simple by keeping a resource allocation graph.
- A simple graph algorithm which detect a cycle in a directed graph can detect a deadlock.
  - Simply find a path from a node to the node itself to find a cycle.

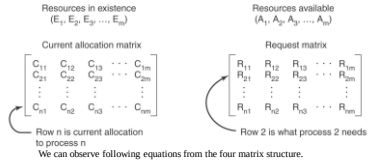
Deadlock Detection and Recovery
(Detection with one resource of each type)



Deadlock Detection and Recovery
(Detection with Multiple resource of each type)

- Operating System need keep matrixes for deadlock detection algorithm.
  - Existing resource matrix (one dimension)– Present how many numbers of resources per each type
  - Available resource matrix (one dimension) – Present how may number of resources per each type are available at any moment, since some of resources are assigned to processes are not available
  - Current allocation matrix (two dimension) – row$_i$ in matrix C present resources currently held by process P$_i$
  - Request matrix (two dimension) – row$_i$ in R present how many more resources are needed for process P$_i$ to finish it's job.

Deadlock Detection and Recovery
(Detection with Multiple resource of each type)



We can observe following equations from the four matrix structure.

$$\sum_{i=1}^{n} C_{i,j} + A_j = E_j$$

Deadlock Detection and Recovery
(Detection with Multiple resource of each type)

- The deadlock detection algorithm is based on comparing vectors.
- Let's define the relation A ≤ B between two one dimension vector A and B means that each element of A is less than equal to corresponding element of B.

Ex)
A = (1, 2, 0 2), B = (1, 3, 0 2): A ≤ B is true? Yes
A = (1, 2, 0, 2), B = (2, 0, 0, 0): A ≤ B is true? No

Deadlock Detection and Recovery
(Detection with Multiple resource of each type)

A Deadlock detection algorithm

1. All process start with unmarked
2. Look for an unmarked process P$_i$, for which the ith row of R (resources needed by P$_i$) is less than or equal to A
3. If such a process is found, add the ith row of C to A, mark the process and go back to step 1
4. If no such a process exists, the algorithm terminate

Deadlock Detection and Recovery
(Detection with Multiple resource of each type)

Ex)
E = (4, 2, 3, 1)
A = (2, 1, 0, 0)

$$C = \begin{bmatrix} 0010 \\ 2001 \\ 0120 \end{bmatrix}, R = \begin{bmatrix} 2001 \\ 1010 \\ 2100 \end{bmatrix}$$

- No deadlock

A = (2, 1, 0, 0)
↓ P$_3$
A = (2, 1, 0, 0)+ (0, 1, 2, 0) = (2, 2, 2, 0)
↓ P$_2$
A = (2, 2, 2, 0)+ (2, 0, 0, 1) = (4, 2, 2, 1)
↓ P$_1$
A = (4, 2, 2, 1)+ (0, 0, 1, 0) = (4, 2, 3, 1)

Deadlock Detection and Recovery
(Detection with Multiple resource of each type)

Ex)
E = (4, 2, 3, 1)
A = (2, 1, 0, 0)

$$C = \begin{bmatrix} 0010 \\ 2001 \\ 0120 \end{bmatrix}, R = \begin{bmatrix} 2001 \\ 2111 \\ 2100 \end{bmatrix}$$

- Deadlock

A = (2, 1, 0, 0)
↓ P$_3$
A = (2, 1, 0, 0)+ (0, 1, 2, 0) = (2, 2, 2, 0)

Deadlock Detection and Recovery
(Detection with Multiple resource of each type)

Ex)
E = (7, 2, 6)
A = (0, 0, 0)

$$C = \begin{bmatrix} 010 \\ 200 \\ 303 \\ 211 \\ 002 \end{bmatrix}, R = \begin{bmatrix} 000 \\ 202 \\ 000 \\ 100 \\ 002 \end{bmatrix}$$

A = (0, 0, 0)
↓ P_1
A = (0, 0, 0)+(0, 1, 0)= (0, 1, 0)
↓ P_3
A = (0, 1, 0)+(3, 0, 3)= (3, 1, 3)
↓ P_2
A = (3, 1, 3)+(2, 0, 0)= (5, 1, 3)
↓ P_4
A = (5, 1, 3)+(2, 1, 1)= (7, 2, 4)
↓ P_5
A = (7, 2, 4)+(0, 0, 2)= (7, 2, 6)

Deadlock Detection and Recovery
(Detection with Multiple resource of each type)

Ex)
E = (7, 2, 6)
A = (0, 0, 0)

$$C = \begin{bmatrix} 010 \\ 200 \\ 303 \\ 211 \\ 002 \end{bmatrix}, R = \begin{bmatrix} 000 \\ 202 \\ 001 \\ 100 \\ 002 \end{bmatrix}$$

A = (0, 0, 0)
↓ P_1
A = (0, 0, 0)+(0, 1, 0)= (0, 1, 0)

Recovery from Deadlock

- Recovery through preemption
  - take a resource from some other process
  - depends on nature of the resource
- Recovery through rollback
  - checkpoint a process periodically
  - use this saved state
  - restart the process if it is found deadlocked

Recovery from Deadlock

- Recovery through killing processes
  - Crudest but simplest way to break a deadlock
  - Kill one of the processes in the deadlock cycle
  - The other processes get its resources
  - The killed process need to start from the beginning