

CSE 120: Principles of Operating Systems

Lecture 6

Deadlocks

October 14, 2003

Prof. Joe Pasquale

Department of Computer Science and Engineering
University of California, San Diego

© 2003 by Joseph Pasquale

1

Before We Begin ...

Read Chapter 8 (on Deadlock)

Midterm is on Tuesday of next week (Oct 21)

- Covers all Lectures including this week
- Covers Programming Assignment #1
- Sample midterm posted on class web page

2

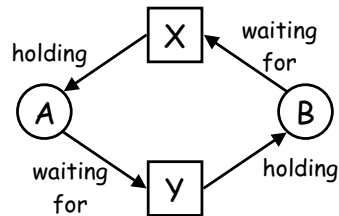
What is Deadlock?

The state of a set of permanently blocked processes

- Unblocking of one relies on progress of another
- But none can make progress!

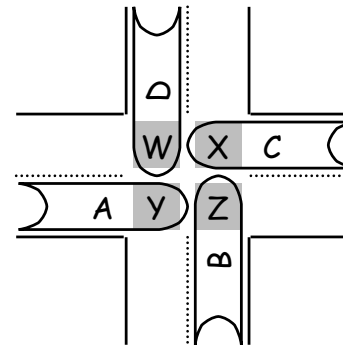
Example

- Processes A and B
- Resources X and Y
- A holding X, waiting for Y
- B holding Y, waiting for X
- Each is waiting for the other; will wait forever

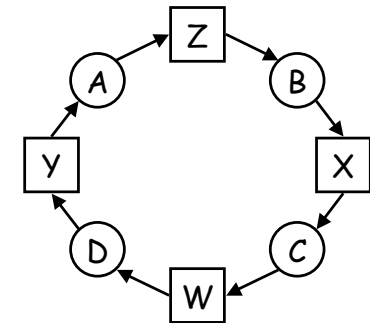


3

Traffic Jam as Example of Deadlock



Cars deadlocked
in an intersection



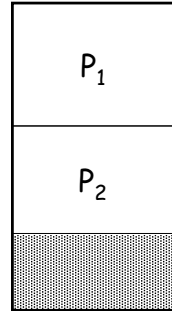
Resource Allocation
Graph

4

More Examples of Deadlock

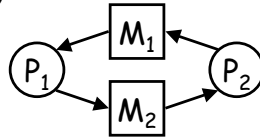
Memory (a reusable resource)

- total memory = 200KB
- P_1 requests 80KB
- P_2 requests 70KB
- P_1 requests 60KB (wait)
- P_2 requests 80KB (wait)



Messages (a consumable resource)

- P_1 : receive M_2 from P_2
- P_2 : receive M_1 from P_1



5

Conditions for Deadlock

Mutual Exclusion

- Only one process may use a resource at a time

Hold-and-Wait

- Process holds resources while waiting for others

No Preemption

- Can't take a resource away from a process

Circular Wait

- The waiting processes form a cycle

6

How to Attack the Deadlock Problem

Deadlock Prevention

- Prevent any possibility of a deadlock

Deadlock Avoidance

- Avoid situations that lead to deadlock

Deadlock Detection

- Don't try to stop deadlocks
- Rather, if they happen, detect and resolve

7

Deadlock Prevention

Simply prevent any one of the conditions for deadlock

- Mutual exclusion
 - Relax where sharing is possible
- Hold-and-wait
 - Get all resources simultaneously, wait until all free
- No preemption
 - Allow resources to be taken away
- Circular wait
 - Order all the resources, force ordered acquisition

8

Preventing a Traffic Jam

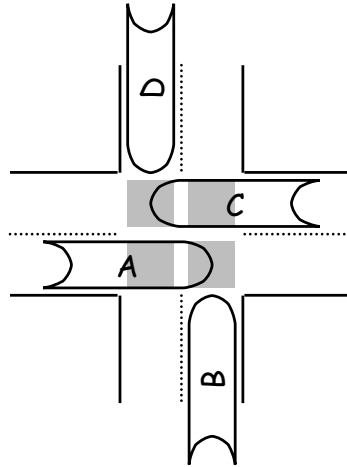
To apply deadlock

prevention to traffic

jam problem, just add

a traffic light!

Which condition is being
prevented?



9

Deadlock Avoidance

Works with *incremental* resource requests

- Resources are asked for in increments
- Do not grant request that can lead to a deadlock

Requires knowledge of maximum resource requirements

10

Banker's Algorithm: Concepts

System has a fixed number of processes and resources

- each process has zero or more resources allocated

System state: either safe or unsafe

- depends on allocation of resources to processes

Safe state

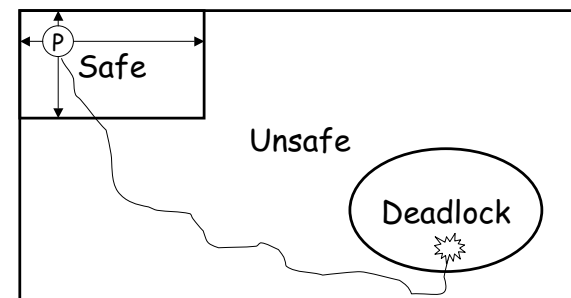
- can avoid deadlock by certain order of execution

Unsafe state

- deadlock is possible (but not necessarily certain)

11

Safe, Unsafe, and Deadlock States



12

Banker's Algorithm

Given

- a process/resource claim matrix
- a process/resource allocation matrix
- a resource availability vector

Is there a sequence of process executions such that

- a process can run to completion, return resources
- resources can then be used by another to complete
- eventually, all the processes complete?

13

Example of a Safe State

Current state

	Claim				Allocation				Avail	Total
	P ₁	P ₂	P ₃	P ₄	P ₁	P ₂	P ₃	P ₄		
R ₁	3	6	3	4	1	6	2	0	0	9
R ₂	2	1	1	2	0	1	1	0	1	3
R ₃	2	3	4	2	0	2	1	2	1	6

This is a Safe State

- Who can run to completion? P₂
- After P₂ completes, it's resources are returned
- Next select P₁, then P₃, then P₄

14

Example of an Unsafe State

Current state

	Claim				Allocation				Avail	Total
	P ₁	P ₂	P ₃	P ₄	P ₁	P ₂	P ₃	P ₄		
R ₁	3	6	3	4	1	5	2	0	1	9
R ₂	2	1	1	2	0	1	1	0	1	3
R ₃	2	3	4	2	0	1	1	2	2	6

So far, this is a safe state, but ...

- if P₂ asks for 1 unit of R₁ and 1 unit of R₃: safe
- if P₁ asks for 1 unit of R₁ and 1 unit of R₃: unsafe
- no one may be able to complete: possible deadlock

15

Avoiding a Traffic Jam

To apply deadlock

avoidance to traffic

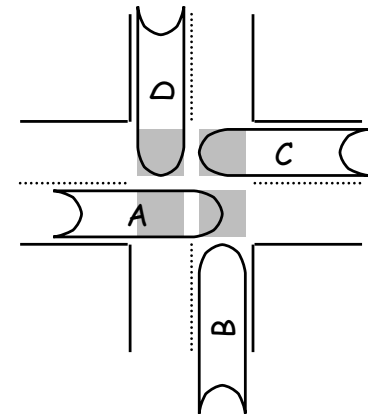
jam problem, allow at

most 3 cars to enter

intersection

What kind of request is

being denied?



16

Deadlock Detection and Recovery

Don't do anything special to prevent or avoid deadlocks

- If they happen, they happen
- Periodically, try to detect if a deadlock occurred
- Do something (or even nothing) about it

Reasoning

- Deadlocks rarely happen
- Cost of prevention or avoidance is not worth it
- Deal with them in special way (may be very costly)

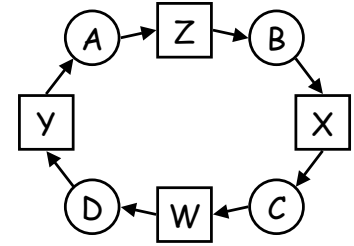
Most systems take this approach!

17

Detecting Deadlocks

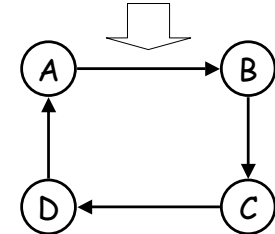
Construct "wait-for" graph

- Construct resource allocation graph
- Remove resource nodes
- If cycle, deadlock



Requires

- identifying all resources and tracking their use
- periodically running detection algorithm



18

Recovery from Deadlock

Abort all deadlocked processes

- Will remove deadlock, but drastic and costly

Abort deadlocked processes one-at-a-time

- Do so until deadlock goes away (need to detect)
- What order should processes be aborted?

What happens to resources in inconsistent states?

- files partially written
- interrupted message (e.g., file) transfer

19