

1. What are the four conditions of deadlock?

Answer:

- Mutual Exclusion
- Hold and Wait
- No preemption on resource
- There is a cycle in the resource request graph.

2. What are some ways to avoid deadlock?

Answer:

- Make all resources shareable
 - Can we do this?
- Don't permit mutual exclusion
 - Can we do this?
- Virtualize non-shared resources
- Use only uniprogramming
 - Run only one process at a time?
- Don't allow waiting and holding
 - Phone company example
 - Process must request everything at once.
 1. How would we know the future?
 2. Prolly request too much
 3. Starvation
 4. May not be possible. . .
- Process must release all current resources before requesting any new ones.
 - Can we do this ALL the time?
 1. Ex: Process that is looking to read something to disk, but can't get disk resource. . .
- Ordered or hierarchical requests
 - Ask for all R1s, R2s, etc. . . Processes must know what they need in advance.
 - One way to do this is using a resource request graph algorithm.

3. A given system has three resources available, R1, R2, and R3. Each resource can accommodate one process and takes some undefined amount of time to complete a request.

Draw a **resource request/allocation graph** of the following sequence of events:

Process A starts and announces it could potentially ask for R1 and R3.
 Process B starts and announces it could potentially ask for R1, R2 and R3.
 Process A asks for R1.
 Process B asks for R2.

Given the graph you just drew, would you want to grant all requests?

Answer:

No, because there would be a cycle in the graph, which would mean possible deadlock!

- One method of preventing deadlock is the Banker's algorithm, which checks to be sure we can allocate resources in a manner that always leaves the system in safe state.

The below system has three types of resources (5 A resources, 10 B resources, and 15 C resources) and three processes. Use the Banker's Algorithm to determine if this system can proceed to a safe state. Is it deadlocked? Explain your reasoning.

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P ₀	2	3	4	3	8	7	1	4	7
P ₁	1	2	3	3	4	5			
P ₂	1	1	1	1	2	9			

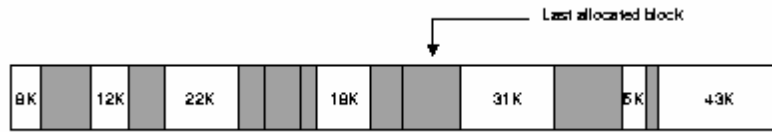
Answer:

	Need		
	A	B	C
P ₀	1	5	3
P ₁	2	2	2
P ₂	0	1	8

As we can see, there's no way to grant any process its maximum resources it requests, so we are in an unsafe state.

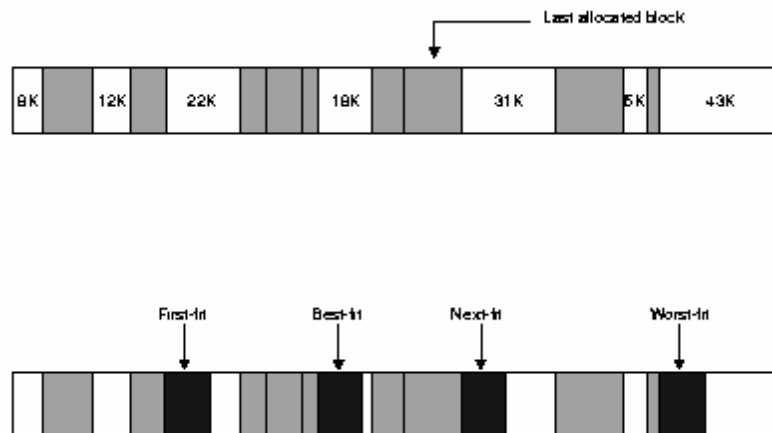
The system is NOT deadlocked yet, however, and no cycle of dependencies exists. If the processes don't request any more resources, they can all complete.

5. The image below give a snapshot of a heap after a program has run for a while. Suppose that the program makes a new() call for a struct that needs 16k of space. Using dynamic storage allocation, show where the block would be placed if *first fit*, *best fit*, and *next fit* policies are used.



Please allocate 16K for the next process.

Answer:



6. What is the difference between internal and external fragmentation?

Answer:

Internal fragmentation – fragmentation within a fixed partition in memory.
 External fragmentation – fragmentation within memory among fixed partitions.