COS Questions – Lecture 4

Operating System Concepts (Tenth Edition)

Deadlocks

8.1 What are the four conditions necessary for a deadlock to occur?

Mutual Exclusion Only one process at a time can access a shared resource.	**Hold & Wait** A process must at least hold and wait an allocated resource while waiting for more.
No Preemption No resources are de-allocated (released) from a process preemptively.	**Circular Wait** A circular chain of processes and resources waiting for each other.

8.2 How does a deadlock present itself in a resource allocation graph?

A cyclic relation typically indicates a deadlock, if all resource instances are used.

8.3 Briefly explain the three strategies for managing deadlocks.

Prevention Guaranteeing that a deadlock will not occur, by assuring that one of the necessary conditions for dead lock is not met.	**Avoidance** Analysis of each new resource request to determine if it could lead to dead lock, and granting it only if dead lock is not possible.	**Detection** Allowing deadlocks, but periodically checking for circular waits, trying to resolve deadlocks if they happen to occur.

8.4 How can a safe state be achieved?

A system is initially in a safe-state. Then, a safe-state is maintained by only performing safe sequences.

8.5 What are the practical problems of recovery from deadlock?

Recovery from deadlock essentially requires either: (a) the termination of a process, or (b) pre-emption of a process's resources. In both cases, a victim process is selected, which is almost always problematic for the end-user.

In typical Windows software, deadlocks are rare enough that it's counterproductive for Windows to deal with them with any strategy more sophisticated than "just let both threads sit there deadlocked until one of the programs gives up or the user manually terminates execution".

8.6 What is the main idea behind the Banker's algorithm?

It's based on the idea that a thread must declare the maximum number of a resource it may need. Then, resources are only granted if it leaves the system in a safe state. Banker's algorithm provides the methods to: (1) determine whether a state is safe (safety algorithm), and (2) whether a request can be granted (Resource-Request Algorithm).

8.7 Given the following snapshot of a system; fill out the matrix $Need$.

	Allocation				Max				Need				Available			
	A	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**
T_0	0	0	1	2	0	0	1	2	0	0	0	0	1	5	2	0
T_1	1	0	0	0	1	7	5	0	0	7	5	0				
T_2	1	3	5	4	2	3	5	6	1	0	0	2				
T_3	0	6	3	2	0	6	5	2	0	0	2	0				
T_4	0	0	1	4	0	6	5	6	0	6	4	2				