Daniel Peterson 109091561

2. Students working at individual PCs in a computer laboratory send their files to be printed by a server that spools the files on its hard disk. Under what conditions may a deadlock occur if the disk space for the print spool is limited? How may the deadlock be avoided?

**A deadlock could occur if many students have their files stored on the hard disk but are not currently printing and a different student is using the printer but does not have space on the hard disk for their files. That student will constantly be requesting memory but will not get it while the other students will be requesting the printer and will not get it. A way to avoid this is to have the student with the printer give it up if they are unable to print.**

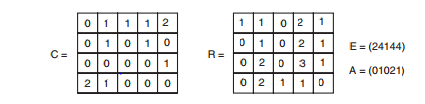
6. City streets are vulnerable to a circular blocking condition called gridlock, in which intersections are blocked by cars that then block cars behind them that then block the cars that are trying to enter the previous intersection, etc. All intersections around a city block are filled with vehicles that block the oncoming traffic in a circular manner. Gridlock is a resource deadlock and a problem in competition synchronization. New York City’s prevention algorithm, called "don’t block the box," prohibits cars from entering an intersection unless the space following the intersection is also available. Which prevention algorithm is this? Can you provide any other prevention algorithms for gridlock?

**This algorithm is an example of hold and wait, which means that a process can only continue if all of the resources it needs are available. This applies to the New York City traffic because cars will wait until they can continue across the entire intersection without blocking traffic. Another prevention algorithm for gridlock would be if cars moved out of the intersection in reverse if they got stuck, but that means that every car behind it would have to move as well to completely clear it.**

8. Is it possible that a resource deadlock involves multiple units of one type and a single unit of another? If so, give an example

**Yes, if two processes are trying to access resources that another is holding then there will be a deadlock. If one process is using all of the units of a given type, and the other process is holding on to the single unit of the other type and each process requests one unit from each other then there will be a deadlock unless one of the processes frees up its resource without getting what it requested.**

14. Consider the following state of a system with four processes, P1, P2, P3, and P4, and five types of resources, RS1, RS2, RS3, RS4, and RS5:



Using the deadlock detection algorithm described in Section 6.4.2, show that there is a deadlock in the system. Identify the processes that are deadlocked

**P1 needs R1 so it cannot run. P2 can run. P3 and P4 need R2 so they cannot run. After P2 runs it returns its resources. P3 can now run and return its resources. P4 now attempts to run and it cannot because it needs R3 and P1 is using R3. P1 can also not run because P4 is using R1 still and cannot run so there is a deadlock with P4 and P1.**

15. Explain how the system can recover from the deadlock in previous problem using (a) recovery through preemption. (b) recovery through rollback. (c) recovery through killing processes.

**a) Take a resource from another process**

**b) Checkpoint a process periodically and use a saved state to restart process if in deadlock**

**c) Kill a process in the deadlock cycle so other processes can get its resources**

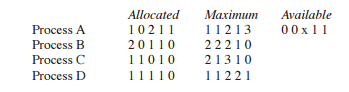
18. Can the resource trajectory scheme of Fig. 6-8 also be used to illustrate the problem of deadlocks with three processes and three resources? If so, how can this be done? If not, why not?

**It can represent 3 processes and 3 resources if the plot is made in 3 directions (x, y and z). The same rules apply and the box where the processes cannot enter is in 3 dimensions, but it is possible.**

25. The banker’s algorithm is being run in a system with m resource classes and n processes. In the limit of large m and n, the number of operations that must be performed to check a state for safety is proportional to m^a \* n^b . What are the values of a and b?

**A = 1 and B = 2. The number of processes must be checked n \* n-1 times which is n^2 and then each process check needs to check each resource (m resources). So we get m \* n \* n-1 which is m \* n^2 or m^1 \* n^2.**

26. A system has four processes and five allocatable resources. The current allocation and maximum needs are as follows:



What is the smallest value of x for which this is a safe state?

**If x is 0:**

**Process A needs 2 of R1 and cannot run. Process B needs 2 of R2 and cannot run. Process C needs 1 of R1 and cannot run. Process D needs 1 of R3 and cannot run.**

**If x is 1:**

**Process A needs 2 of R1 and cannot run. Process B needs 2 of R2 and cannot run. Process C needs 1 of R1 and cannot run. Process D gets the R3 and R4 it needs and executes and returns its resources. Then process A still cannot run because it needs 1 more of R1. Process B needs 1 more of R2 and cannot run. Process C still needs 1 more of R3 and cannot run.**

**If x is 2:**

**Process A needs 2 of R1 and cannot run. Process B needs 2 of R2 and cannot run. Process C needs 1 of R1 and cannot run. Process D gets the R3 and R4 it needs and executes and returns its resources. Then process A still cannot run because it needs 1 more of R1. Process B needs 1 more of R2 and cannot run. Process C gets R3 and runs and returns its resources. Then processes A and B can run to completion.**

**The smallest value of x for a deadlock free state is x = 2.**

28. Two processes, A and B, each need three records, 1, 2, and 3, in a database. If A asks for them in the order 1, 2, 3, and B asks for them in the same order, deadlock is not possible. However, if B asks for them in the order 3, 2, 1, then deadlock is possible. With three resources, there are 3! or six possible combinations in which each process can request them. What fraction of all the combinations is guaranteed to be deadlock free?

**If both processes ask for the same record then one will be blocked and no lock will occur. If one asks for a resource and the other asks for a different resource then they will both get the resource and neither will complete unless the other gives up the resource they just got. Both combinations for process B which request resource 3 first (321 and 312) or resource 2 first (231 and 213) will deadlock because the other process will never give up its resource. Both combinations which the first resource is requested first (123 and 132) will not deadlock because the other process will be blocked and no deadlock will occur because it will not request the other resources.**

34. Explain the differences between deadlock, livelock, and starvation.

**Deadlock and livelock are types of starvation. Starvation is a lack of forward progress meaning that the process can no longer get resources and cannot execute. Deadlock is also a lack of forward progress which will never resolve because it involves a cyclic dependency with one process to another. Each process needs the other to complete and the cycle cannot break. Livelock as well is a lack of forward progress but it can break. Normally it involves high priority jobs taking resources away from low priority jobs in order to complete and break the lock.**