King Saud University
Department of Computer Science
CSC227:Operating Systems
Tutorial No. 7

1) **Describe the difference between deadlock and starvation.**

The difference between deadlock and starvation is that with deadlock none of the threads in a set of threads are able to make progress because the events they are waiting for can only be triggered by other threads in the set that are also blocked. With starvation, some threads may make progress while others fail to ever make progress because for example, they are losing in all races for a semaphore. As another example, if a LIFO queue is used for locks, some threads may starve if they happen to end up at the bottom of the queue

2) **Is it possible to have a deadlock involving only one single process? Explain your answer.**

No, because of the hold-and-wait condition. If you are holding resources, you are not waiting for them.

3) **Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock- free.**

Suppose the system is deadlocked. This implies that each process is holding one resource and is waiting for one more. Since there are three processes and four resources, one process must be able to obtain two resources. This process requires no more resources and, therefore it will return its resources when done.

4) **Can a system detect that some of its processes are starving? If you answer yes, explain how it can. If you answer no, explain how the system can deal with the starvation problem.**

No. A process starves if it never gets to run. If a given process has not run for 1 second, has it starved? How about if it has not run for 10 seconds? A minute? An hour? None of these indicate that a process is starving, since it may get to run in the next second. However, as the amount of time that a process has waited gets longer and longer, the probability that it is starving goes up.
This argument depends on the fact that no numeric criteria exist for declaring a process to have starved. If, on the other hand, we declared a process to have starved if it waits 10 seconds without ruining, then we might be able to answer "yes".

5) **What is the meaning of the term busy waiting? What other kinds of waiting are there in an operating system? Can busy waiting be avoided altogether? Explain your answer.**

Busy waiting means that a process is waiting for a condition to be satisfied in a tight loop without relinquish the processor. Alternatively, a process could wait by relinquishing the

processor, and block on a condition and wait to be awakened at some appropriate time in the
future. Busy waiting can be avoided but incurs the overhead associated with putting a process to
sleep and having to wake it up when the appropriate program state is reached.