Chapter 6 - Deadlock and Starvation (Quiz)

1.	**The _____ allows multiple threads to have simultaneous read-only access to an object protected by the lock.**	readers/writer lock
2.	**A closed chain of processes exists, such that each process holds at least one resource needed by the next process in the chain is the condition of _____ .**	circular wait
3.	**The _____ condition can be prevented by defining a linear ordering of resource types.**	circular wait
4.	**The _____ condition can be prevented by requiring that a process request all of its required resources at one time and blocking the process until all requests can be granted simultaneously.**	hold and wait
5.	**Examples of _____ include processors, I/O channels, main and secondary memory, devices, and data structures such as files, databases, and semaphores.**	reusable resources
6.	**The fastest form of interprocess communication provided in UNIX is _____ .**	shared memory
7.	**A _____ is a software mechanism that informs a process of the occurrence of asynchronous events.**	Signal
8.	**The _____ is useful in sending a signal to a thread indicating that a particular event has occurred.**	event object
9.	**The most common technique used for protecting a critical section in Linux is the _____ .**	spinlock
10.	**Once the processes have progressed into the _____ , those processes will deadlock.**	fatal region
11.	**One of the most significant contributions of UNIX to the development of operating systems is the _____ .**	pipe
12.	**Requested resources are granted to processes whenever possible with _____ .**	deadlock detection
13.	**A set of processes is _____ when each process in the set is blocked awaiting an event that can only be triggered by another blocked process in the set.**	Deadlocked
14.	**The strategy of deadlock _____ is to design a system in such a way that the possibility of deadlock is excluded.**	prevention
15.	**With _____ only one process may use a resource at a time and no process may access a resource unit that has been allocated to another process.**	mutual exclusion