1.  What is a process and what is the four principal ways they are created?

A process is an instance of a computer program that the OS gives resources to accomplish tasks. There are four principles for how they are created including system initialization, an execution of system processes, a user’s request to create a new process and an initiation of a batch job.

2.  What is a daemon?

Daemon is a multitasking operating system that is build for running certain processes in the background (things that the user can’t see) so that the operating system can utilize resources effectively while keeping the GUI clean.

3.  How can processes be terminated?

A process can either be terminated by completing its assigned tasks and then terminating or by being “killed.” The user or the operating system can choose to kill a process early if it’s no longer needed.

4.  Explain the three process states?

Processing states is a useful abstraction for understanding how a kernel deals with using processes. There are five major states: created, running, waiting, blocked and terminated. Each one of these is fairly self explanatory but these states allow the operating system to organize tasks so the processor can be used to the greatest extent.

5. What is a process table and what type of information does it contain?

A process table is a table that stores information about the current processes that are in the computer. These include process id, process owner, process priority, environment variables, parent processes, and pointers to executable.

6.  Explain the concept of multithreading and give an example.

Multithreading is the ability for an operating system to run tasks simultaneously in order to create OS efficiency. An example would be the process of Microsoft word, within this process there are multiple threads working at the same time. As I type there is a thread for spell check, grammar, printing the words to the screen as I type, etc.

7.  Explain mutual exclusion and critical region, and how they are related.

This process is to ensure that no two processes are within their critical regions at the same time. A critical region would be a shared resource like shared memory. If two processes attempted to remove or modify data at the same time, pointers or programming referencing them could be severely messed up.

8.  Compare and contrast semaphores, monitors, and message passing.

These all have to do with multiprocessing and synchronization across an operating system. A semaphore is like a record that the operating system can check with to see how much of a resource is available. A monitor allows for mutual exclusion as well as the ability to wait for things to be freed up. The monitor is different from a semaphore because it is the way the program waits. The last is message passing which is used as a form of object communication where the different threads and programs can talk with each other to create their own form of queue.

9.  Compare and contrast the scheduling algorithms used in batch systems.

A few scheduling algorithms include First-in-First-Out (FIFO), shortest job first, shortest remaining time, and highest response. FIFO is what we talked about last week with the queue actin like a line where the first person in is the first one serviced. This would be the same with the processes where they would be executed depending on their arrival. Shortest remaining time is self explanatory but requires an understanding that the operating system can make accurate calculations of expected times. I’ve read that this can be very problematic with open source Os’ because they programs will often have unfamiliar process so this style will make massive mistakes. Highest response judges how many of the threads are outputting useful executions and allocates resources based on productivity; same problems and advantages.

10. What is a real-time system, and what are the two types?

There are Soft real time systems and Hard real time systems. A soft system is one that partially meet the deadlines for execution so that it can focus on the round deadline while a hard system is strict with it’s time.

11. What is the difference between turnaround time and response time?

A turnaround time is measured from it’s submission to it’s completion while response time is from the request to the first response.

12. What is a race condition?

A race condition is exactly the situation that you try to avoid with some of the algorithms that were mentioned earlier. This occurs when two or more threads try to access or change data at the same time and whatever algorithm that you’ve chosen must decide the order.