1.  Explain the purpose of the base and limit registers in a multiprogramming operating system?

A base register is loaded with the physical location where the process begins in memory while the limit register is loaded with the length of it. For a multiprogramming operating system to function efficiently one would need both of these to accurately map processes without the processes overwriting each other.

2.  Explain swapping as it applies to virtual memory?

Your operating system will create another file on your hard disk for paging that will trick the program that it’s running into thinking it has more memory available than it actually does. This will usually occur because you’ve run out RAM.

3.  Explain the memory management algorithms discussed in Section 3.2.3?

The first algorithm is the First-In-First-Out method which does exactly what it sounds like in terms of paging: it handles the pages in the first come first serve basis. In contrast there is the Last-In-First-Out method, which does the opposite and handles the latest pages first. The Working set looks at the pages that are requested by the operation and works on the assumption that the program will not call all of them which would cause it to crash.

4.  Explain the difference between pages and segments?

Memory paging is used to store memory that would not fit onto the RAM to the secondary device like a hard drive and keep them ready to use. Segmenting is the operating systems way of sectioning off areas of memory of a specific size and ensuring that new processes do not overwrite the old ones.

5. What is a "dirty" bit, and how does it affect performance in paging system?

A dirty bit is set if the page or memory has been modified in any way and is used to indicate that there have been changes.

6.  What is a Translation Lookaside Buffer used for?

This is used as a cache that the memory management software will use to improve virtual address translation speed. This will look at the table with all of the locations of the pages and look for the next ones that are coming up and preload them.

7.  Compare and contrast Not Recently Used and First-In, First-Out replacement algorithms.

The similarities between these two algorithms are mathematical with their equations for competitive complexity being equal. These two tend to have similar results due to the way pages are referenced and modified. The difference comes when you look at how the Not Recently Used (NRU) prioritizes pages. In NRU, the pages are marked as modified and/or referenced and are given priorities based on whether they are being referenced or modified

8.  Explain locality of reference and does it have an impact on cache  memory?

This is known to describe an area that is frequently visited in memory that, in terms of cache memory, offers a insight into the most active areas of the memory.

9.  Explain thrashing.

Thrashing is when the computer’s virtual memory is rapidly paging it’s virtual memory to the disk which causes a huge degradation in performance.

10.  Explain the difference between internal and external fragmentation.

Internal fragmentation occurs when a program requests a certain amount of memory but doesn’t use all of it. For instance, if a program needs 23 bits of data for storage the processor will give it 24 because that’s how it allocates chunks of memory (bits, bytes, etc). External fragmentation occurs when the memory allocation software allocates memory for programs that is separated by small amounts of space that eventually add up to a useable amount of data. At this point that memory cannot be used due to its size.

Chapter 3 Problems:

4. 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.

18. 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.

28. 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.