Week 7 Topics:
Operating System Details: Tasks, Processes, and Memory management

As you saw from the Midterm Exam, these questions actually DID form the basis of the questions I used for the exam. Didn't believe me, did you? Now that you know this, why not study these now and review them weekly with me on Mondays! I may actually ask you to do something with these in class!

1. What is the difference between a program and a process?
   A program is a passive (resides on disk) entity that consists of one or more processes.
   A process is an active (resides in memory) entity that consists of one or more threads.

Core was "Singed" - probably from over-clocking!

2. How many instructions can a singe core of a processor be executing at any instant?
   One - We'll discuss "superscalar", "scalar", "pipelining" and other speed/time compression techniques in COMP 183

3. What are the five possible states that a process can be in?
   Hold/Requested - Process has been called and awaiting to be granted access to the ready state by the scheduler.
   Ready - Process has been loaded into memory and is awaiting its turn with the CPU.
   Running - Process has control of the CPU and is performing the machine cycle at its most basic level.
   Waiting - The process is blocked, waiting on something else, or delayed and not occupying the CPU.
   Finished - The process has completed it's task, is released from the CPU queue, and returned to the user.

4. What causes them to transition from one state to another? The three levels of the "Scheduler" software
   Long-term - Controls entry and exit from the CPU queue
   Medium-term - Controls swapping between active processes
   Short-Term - Handles the "housekeeping" at the CPU level (More in COMP 183 - stay tuned!)

5. When a process is sleeping, what causes it to stop sleeping? An "interrupt" caused by a I/O event or signal, or by pre-set software timer.

O(1) older

6. Describe the common scheduling algorithm that Windows and Linux use: The "Multilevel Feedback Queue" is a layered approach to scheduling. It employs a handful of FIFO (first in first out) queues with increasingly longer time quanta (time-slices to complete a task) with a base "round-robin" or FCFS (First Come, First Served) queue. Processes move between queues based on their ability to complete their actions within the quantum assigned - down if it doesn't, up if it becomes I/O bound, same if it plays "nice".

7. Describe, briefly what each of the following Linux Process states means:
   (a) D uninterruptible sleep (usually IO) Sleep state is on a timer and will proceed only after the timer has run out.
   (b) R runnable (on run queue) Process is prepared to use the CPU
   (c) S sleeping Process is waiting on a resource
   (d) T traced or stopped Process is suspended and cannot execute
   (e) Z a defunct ("zombie") process The process is waiting to terminate but cannot for some reason. Usually as a result of it's parent process having ended without sending the process a signal to also terminate.

8. What are some problems with the following scheduling algorithms:
   (a) First come first serve Processes can get caught behind a single process that may be "bogarting" the processor. (EG: A process is performing I/O and will not relinquish the CPU while it waits.)

   (b) Shortest task first Long tasks may not get the chance to run if the Processor has been handed many short tasks.

   (c) Highest priority first Very-low or low priority tasks can become stuck in the queue in perpetuity.

9. What is the order of magnitude O(?) of the new "CFS" scheduling algorithm in the Linux 2.6 kernel? (*The answer is in the Linux 2.6 Kernel Scheduling "Information here" link in the Process Scheduling lecture page.*) O(log n) where "n" is the number of nodes in the "Red-Black" binary tree structure A "red-black" tree is self-balancing binary tree algorithm that ensures that the "depth" of each branch is essentially uniform across the entire tree.

10. How many different priorities can there be in the 2.6 Kernel? 140 Queues
    What is significance of tasks less than 100? (Note the lower the number, the higher the priority.) These were (are) reserved for "realtime" processes

11. What current (It is – Albeit Open Source) operating system uses the simple "Single Fixed Partition" memory management scheme? DOS

12. What is the "Multiple Fixed Partitions, Fixed Tasks" memory management scheme? (Old!) Where memory allocations are pre-defined and reserved for specific classes or types of program, and programs are compiled to only store themselves in the appropriate memory class allocations.
    What is it's advantage over the Single Fixed Partition scheme? More efficient use of memory and allows for swapping in and out of programs.
    What are several disadvantages? Each partition will require a job queue, programmer needs to know which queue to develop for ahead of time, potential for large swaths of memory to be un/under-utilized, partitions are defined when the operating system is built. Memory expansion becomes more problematic (recompile the kernel).

13. When we go to a "Multiple Fixed Partitions, Movable Tasks" memory management scheme, what is the main functionality that has to be added? Relocatable loader software or equivalent hardware (base registers) that re-map the task from its "preferred" address location to its actual address location.

14. What is the main advantage of "Multiple Fixed Partitions, Movable Tasks" memory management scheme? If one partition is "busy", then the process can be loaded into another partition without having to wait for the preferred partition to come free or swap out.

15. Describe the "Multiple Variable Partitions" memory management scheme: Each partition is alocated at "run-time" (when the program requests it) to perfectly fit any given process' needs.

16. What are two advantages and two disadvantages of the “Multiple Variable Partitions” memory management scheme?

(Advantage)	(Disadvantage)
Allows for many programs to be loaded at once with efficient use of memory space (less internal fragmentation)	Requires sophisticated hardware or loader software
Dynamic and swap-friendly with no memory boundaries	Becomes more inefficient with larger processes and greater load on resources.

17. What are the main advantages of “Paged Memory Systems”? Low fragmentation (both Internal and external, though internal is more than external) and no need for memory compaction.

How are these advantages accomplished? Each allocation unit is of a fixed size (often 4096 bytes) and the use of modern (i386 or newer) hardware registers allowing each process to "think" it is loaded starting at byte "0x000"

18. What is the biggest difference between the “Paged Memory System” and “Paged Virtual Memory System”? That the Paged Virtual Memory system only loads pages of memory as needed when and appropriate "page fault" memory interrupt occurs.

19. On a 32 bit processor, how much memory does an application program running on a Virtual Memory System "think" it has available to it? 4 GB

20. What is a working set? This is the amount of memory that a given process "needs" in any given moment in time.

21. What is Physical Address Extension? It is an addition of more "lines" into the physical design of a processor that can increase and map the maximum possible physical memory size beyond that of virtual address space.

Why might this be important to know? Physical memory is "faster" to access than virtual memory. Which is why when dealing with server hardware, they often have more physical RAM than address space given the amount of work they must do, and this extra memory needs to be accessed by all processors and processor cores, somehow.

22. What does segmentation add to the Virtual memory management scheme? _____

23. Describe internal and external fragmentation. Internal fragmentation is where a process is assigned more memory resources than it needs. External fragmentation is where memory cannot be allocated for a given process because there isn't a "slot" or "hole" in physical memory big enough to allocate a contiguous amount for the process' working set.

24. What is thrashing? Where a system is paging in and out constantly to the point where the paging is taking over as the primary consumer of system resources.

What causes it? A lack of system resources such as insufficient memory, or poor software design where the working set of an application exceeds the capabilities of the system, or even improper configuration of the virtual memory on the system.

25. How much virtual and physical memory can an X86-64 system have? 256 TB (2^48 Bytes)

26. What is the limitation on physical memory size? On an x64 system its cost and "real-estate". Imagine, if you will, what it would cost to populate a computer with 256 TB of SDRAM chips (currently at $9/GB) - $3 Million! AND… How big would that case have to be to fit 1000, 256 GB DIMM (Dual Inline Memory Modules) in it?