1. (Chapter 1) What are the three main purposes of an operating system?
   1. Run programs in a convenient and efficient manner
   2. Manages hardware
   3. Allocates resources
2. (Chapter 1) Explain the purpose of an interrupt vector.
   1. Points to the location of the interrupt process
3. (Chapter 1) What is a bootstrap program and where is it generally stored?
   1. Program that is run on startup and begins the rest of the programs
   2. ROM hardware read only memory
4. (Chapter 1) What role do device controllers and device drivers play in a computer system?
   1. They handle the interactions between a device and the OS
5. (Chapter 1) Why are clustered systems considered to provide high availability service?
   1. If one system is down the whole cluster can continue running
6. (Chapter 1) Describe the differences between physical, virtual and logical memory?
   1. The memory where everything is stored
   2. Ram
   3. resectioning and renumbering of running memory
7. (Chapter 1) Describe the differences between kernel mode and user mode?
   1. Kernel has direct access to all the processes and drivers
   2. Goes through levels of separation to handle more complicated processes
8. (Chapter 1) What two operating systems currently dominate mobile computing?
9. iOS & Android
10. (Chapter 2) Describe the relationship between an API (e.g. Posix), the system call interface and the operating system?
    1. The SCI stands between the API and OS. It handles commands and translation between the two
11. (Chapter 2) Describe three general methods used to pass parameters to the operating system during system calls?
    1. Registers: hold the data and send it one at a time in
    2. Blocks: a collection of data points sent at once
    3. Stack: first in last out data
12. (Chapter 2) What is a microkernel?
    1. A kernel striped down to its essential pieces to run
13. (Chapter 3) Name and describe the different states that a process can exist in at any given time.
    1. New, ready, running, waiting, terminated
14. (Chapter 3) Explain the differences between an I/O bound-process and a CPU-bound process.
    1. Uses more input output than computation
    2. The opposite
15. (Chapter 3) Explain the concept of a context switch and what happens during a context switch.
    1. Moving from one process to another
16. (Chapter 4) Why should a web server not run as a single threaded process?
    1. Multiple users will require multiple processes and one thread wouldn’t handle that
17. (Chapter 4) Describe what deferred cancellation of a thread is and how it can ensure that thread termination occurs in an orderly manner as compared to asynchronous cancellation.
    1. Making sure the program is ready to shut down. Sending the command and letting the program shut itself down
18. (Chapter 4) What is a thread pool and why is it used?
    1. Multiple threads just waiting to be used for any process that needs one
19. (Chapter 4) Distinguish between parallelism and concurrency.
    1. Multiple threads running programs at the same time
    2. Single linear one at a time programs
20. (Chapter 4) Describe how OpenMP is a form of implicit threading.
    1. It creates threads for programs that need them.
21. Chapter 5) What three conditions must be satisfied in order to solve the critical section problem? Describe each of the three conditions.
    1. Mutual exclusion: Only one program in critical section at once
    2. Progress: programs that are waiting decide which will enter critical next
    3. Bounded waiting: Time between wanting access and being granded
22. (Chapter 5) Describe the difference between a preemptive and a nonpreemptive kernel.
    1. Allows and doesn’t allow programs to be preempted
23. (Chapter 5) Describe how either the test\_and\_set atomic instructions can be used as the basis to solve the critical-section problem using a short C pseudo code example.
    1. boolean test and set(boolean \*target) {
       1. boolean rv = \*target;
       2. \*target = true;
       3. return rv;
    2. }
24. (Chapter 5) Explain the difference between the first readers-writers problem and the second readers-writers problem.
25. (Chapter 5) Describe how a counting semaphore works. You may use pseudo code as part of your description.
    1. It is a counting value that is shared between programs
26. (Chapter 5) What data structure can be added to the classic semaphore implementation to improve its performance?
    1. struct
27. (Chapter 5) How is the priority inversion problem typically solved?
    1. Priority-inheritance protocol
28. (Chapter 5) The “synchronized” method in Java is an example of what synchronization mechanism being incorporated into Java?
    1. monitor-like concurrency mechanism
29. (Chapter 6) Explain the concept of a CPU–I/O burst cycle.
    1. Swapping between processing and file manipulation
30. (Chapter 6) What effect does the size of the time quantum have on the performance of an Round Robin CPU scheduling algorithm?
    1. If it is too small the overhead becomes too high
31. (Chapter 6) Explain the process of starvation and how aging can be used to prevent it in a priority scheduling algorithm for the CPU.
    1. Low level programs never being executed
    2. Longer wait times raise priority
32. (Chapter 6) Describe two general approaches to load balancing for multiple-processor scheduling.
    1. Push migration: pushed program from full core to not full core
    2. Pull migration: idle processors pull tasks from busy ones
33. (Chapter 6) Describe the differences between soft and hard processor affinity.
    1. Process can move to different processors
    2. Process specifies processor
34. (Chapter 6) What are the advantages of the Earliest-Deadline-First scheduling algorithm over the rate-monotonic scheduling algorithm?
    1. Optimal for meeting deadlines
35. (Chapter 7) Explain what must happen for a set of processes to achieve a deadlocked state.
    1. Two process are waiting on each other for a signal to continue
36. (Chapter 7) Describe the four conditions that must hold simultaneously in a system if a deadlock is to occur. Include more than just the names of the conditions.
    1. Mutual exclusion: only one process at a time can use a resource
    2. Hold and wait: a process holding at least one resource is waiting to acquire additional resources held by other processes
    3. No preemption: a resource can be released only voluntarily by the process holding it, after that process has completed its task
    4. Circular wait: there exists a set {P0, P1, …, Pn} of waiting processes such that P0 is waiting for a resource that is held by P1, P1 is waiting for a resource that is held by P2, …, Pn–1 is waiting for a resource that is held by Pn, and Pn is waiting for a resource that is held by P0.
37. (Chapter 7) What are the three general ways that a deadlock can be handled, and which one is used by most operating systems?
    1. Ensure there will never be a deadlock
    2. Allow the system to enter, detect it, then resolve
    3. Ignore the issue and continue (Most Common)
38. (Chapter 8) What is the advantage of using dynamic loading of memory?
    1. A routine is not loaded until it is called. Only its route is stored
    2. Does not require special support from OS
39. (Chapter 8) Distinguish between internal and external memory fragmentation.
    1. When there is more space assigned than needed
    2. When there is enough space but it is broken into separate pieces
40. (Chapter 8) Explain the basic method for implementing paging.
    1. Breaking physical and logical memory into blocks and keeping track of which block references the other. Page and frame (logical, physical)
41. (Chapter 8) Describe how a transaction look-aside buffer (TLB) assists in the translation of a logical address to a physical address.
    1. It keeps a select number of transitions loaded to speed up the process of finding the frame
42. (Chapter 8) Briefly describe the segmentation memory management scheme. How does it differ from the paging memory management scheme in terms of the user's view of memory?
    1. Half of the memory is kept hidden and private to a certain process
43. (Chapter 9) Explain the sequence of events that happens when a page-fault occurs in a virtual memory system.
    1. The page is requested
    2. If the page is invalid a trap is sent to the OS
    3. A free frame is found
    4. New frame is read into the physical memory
    5. Page table is reset to contain new page
    6. Instruction is reset with page loaded
44. (Chapter 9) Explain the Enhanced Second-Chance Algorithm and the usefulness of the modify bit in this algorithm.
    1. Improved algorithm that uses reference and modify bit
    2. It says whether or not the page needs to be written to memory before it can be replaced