1. How do **processor affinity** and **load balancing** interact with each other in a multi-processor system?
2. Improving load balancing always satisfies the processor affinity requirement.
3. Processor affinity and load balancing conflict with each other in all cases.
4. Moving processes between processors to improve load balancing may conflict with processor affinity.
5. Processor affinity and load balancing are totally unrelated and can be handled independently.
6. Which of the following is **not** true about the different solutions to dynamic storage allocation?
7. First fit is faster than best fit and worst fit.
8. Best fit has the advantage of producing the smallest leftover hole.
9. Worst fit has the advantage of producing the largest leftover hole.
10. Best fit is better than first fit in all cases.
11. Handling a page fault involves many steps. Which step is the most time consuming?
12. Identifying the interrupt as a page fault.
13. Saving the context of the process that caused the page fault.
14. Transferring data from disk to memory.
15. Updating the page table to indicate that the missing frame is now in physical memory.
16. In a certain system, the TLB access time is 1 ns, and memory access time is 200 ns. If the TLB hit ratio is 0.9, what’s effective memory-access time?
17. 201 ns b. 220 ns c. 380 ns d. 180 ns e. 290 ns
18. Which of the following is true about paging?
19. Paging eliminates internal fragmentation but does not eliminate external fragmentation.
20. Paging eliminates external fragmentation but does not eliminate internal fragmentation.
21. Paging eliminates both external fragmentation and internal fragmentation.
22. Paging does not eliminate any kind of fragmentation.
23. If a process uses 5000B in a system with a page size of 1KB, what’s the size of internal fragmentation?
24. 120B b. 904B c. 1KB d. 96B e. 0B
25. How does page size affect memory utilization?
26. A smaller page size decreases internal fragmentation and increases the page table size.
27. A smaller page size decreases internal fragmentation and decreases the page table size.
28. A smaller page size increases external fragmentation and increases the page table size.
29. A smaller page size decreases internal fragmentation but does not affect the page table size.
30. What is the difference between the **Scan** and the **Look** disk scheduling algorithms?
31. **Scan** moves the head before rotating the disk, while **Look** does the opposite.
32. **Scan** moves from one end of the disk to the other end, while **Look** always starts from the middle.
33. **Scan** always moves from one end of the disk to another, while **Look** does not reach a disk end unless there is a request at that end.
34. **Scan** minimizes seek time, while **Look** minimizes rotational time.
35. **Scan** always results in a smaller average seek time than **Look**.
36. Which disk scheduling algorithm is the most likely algorithm to cause starvation?

a. Scan b. C-Scan c. Look d. C-Look e. SSTF f. FCFS

1. Adding more frames may cause more page faults if the following page replacement algorithm(s) is(are) used:
2. FIFO b. Optimal c. LRU d. Both FIFO and LRU e. Both LRU and Optimal
3. Which of the following is **not** true about memory frame allocation?
4. The number of frames allocated to a process is always proportional to its size.
5. If global frame allocation is used, a process’s execution time will depend on other processes.
6. Local frame allocation may result in poor memory utilization.
7. The minimum number of frames allocated to a process is hardware dependent.
8. What’s the relationship between the working set size, page fault rate and thrashing?
9. Larger working set sizes lead to higher page fault rates, thus increasing the chance of thrashing.
10. Larger working set sizes lead to lower page fault rates, thus reducing the chance of thrashing.
11. Thrashing depends on the page fault rate and has nothing to do with the working set size.
12. Neither the working set size nor the page fault rate can affect thrashing.
13. Which of the following is **not** true about a **Zombie** process?
14. It has an entry in the process table.
15. Its parent has terminated without calling wait().
16. Its parent has not called wait(), but the parent has not terminated yet.
17. Its resources have been deallocated by the operating system.

14. Which of the following is **not** true about virtual memory (VM)?

1. With VM, a program can be run even if the size of its address space exceeds the amount of physical memory.
2. Using VM increases the amount of parallel programming.
3. A VM system never loads into physical memory an *instruction* that the program will not execute.
4. Implementing VM requires hardware support.

15. Which of the following is true about segmentation and paging?

1. Segmentation requires hardware support but paging does not.
2. Paging divides a process’s address space into equal blocks, while segmentation may divide the address space into unequal blocks.
3. The different pages of a process must be adjacent in memory while the segments of a process may be scattered all over memory.
4. The different segments of a process must be adjacent in memory while the pages of a process may be scattered all over memory.
5. Both b and c are true.
6. Both b and d are true.
7. Which of the following events will cause the OS to put a process in the waiting state?
8. Requesting an I/O operation.
9. Completing an I/O operation.

c. Executing a memory access that causes a page fault.

d. Expiration of the process’s time quantum.

e. Both b and c are correct. f. Both a and c are correct. g. Both a and b are correct.

1. Which of the following is true about threads and processes?
2. Threads within a process share the same stack.
3. Threads within a process share global variables.
4. Threads within a process must run on the same core.
5. Context switching between processes is faster than context switching between threads.
6. Both b and d are correct. f. Both a and b are correct. g. Both b and c are correct.
7. Which of the following is (are) true about disk scheduling algorithms?
8. Scan always reaches the end of the disk, while C-Scan may not reach a disk end.
9. Scan always reaches the end of the disk, while Look may not reach a disk end.
10. Scan provides a more uniform waiting time than C-Scan.
11. FCFS is likely to result in too many head movements compared to other algorithms.
12. Both b and d are correct. f. Both b and c are correct. g. Both c and d are correct.
13. Which of the following is true about waiting time in disk scheduling?
14. FCFS always gives bounded waiting time.
15. SSTF does not guarantee any bound on the waiting time.
16. Scan does not guarantee any bound on the waiting time.
17. Look does not guarantee any bound on the waiting time.
18. Both a and d are correct. f. Both a and c are correct. g. Both a and b are correct.
19. Which of the following is true about paging?
20. Paging eliminates internal fragmentation but does not eliminate external fragmentation.
21. Paging eliminates external fragmentation but does not eliminate internal fragmentation.
22. Paging eliminates both external fragmentation and internal fragmentation.
23. Paging only reduces fragmentation but does not eliminate any kind of fragmentation.
24. Paging does not affect fragmentation.
25. Which of the following is true about paging and segmentation?
26. Paging divides memory into unequal blocks.
27. Segmentation always divides memory into equal blocks.
28. A single instruction may access multiple pages.
29. A single operand may access multiple pages.
30. Both a and c are correct. f. Both b and d are correct. g. Both c and d are correct.
31. A logical address space of a process has 256 pages with a 4-KB page size. How many bits are required in the logical address?

a. 8 b. 12 c. 16 d. 20 e. 24 f. 32 g. 40 h. 64

1. If a process uses 4000B in a system with a page size of 1KB (1024B), what’s the size of internal fragmentation?
2. 24B b. 96B c. 904B d. 1KB e. 0B
3. Which of the following **is not necessarily** performed by the kernel in handling a page fault?
4. Issuing a read request to the disk to fetch the missing frame into memory.
5. Saving the state of the process that caused the page fault.
6. Granting the CPU to a process other than the process that caused the page fault.
7. Updating the page table to indicate that the missing frame is now in physical memory.
8. Running an interrupt service routine.

1. Which of the following scheduling strategies can minimize the response time of interactive processes?
2. Lowering the priority of a process after it is released from waiting for I/O.
3. Boosting the priority of a process after it is released from waiting for I/O.
4. Boosting the priority of a process that does not complete within its time quantum.
5. Boosting the priority of a process that does not use its entire time quantum.
6. Both b and d are correct. f. Both a and c are correct. g. Both b and c are correct.
7. Adding more frames may cause more page faults if the following page replacement algorithm(s) is (are) used:
8. FIFO b. Optimal c. LRU d. Both FIFO and LRU e. Both LRU and Optimal
9. Which of the following is (are) true about the working set size?
10. A larger working set size increases the chances of having page faults.
11. A larger working set size reduces the chances of having page faults.
12. If the sum of working set sizes exceeds the number of available frames, the system is thrashing.
13. The working set size of a process is constant.
14. Both b and c are correct. f. Both a and d are correct. g. both a and c are correct.
15. How does contiguous disk allocation compare with indexed allocation?
16. Contiguous provides faster random access, while indexed provides faster sequential access.
17. Contiguous provides faster sequential access, while indexed provides faster random access.
18. Contiguous provides faster sequential access but causes external fragmentation.
19. Contiguous provides faster random access and less external fragmentation.
20. Both b and c are correct. f. Both c and d are correct. g. Both a and c are correct.

1. How does adding a File Allocation Table (FAT) improve linked disk allocation?
2. It provides faster random access.
3. It provides faster sequential access.
4. It reduces external fragmentation.
5. It reduces internal fragmentation.
6. Both a and b are correct. f. Both a and c are correct. g. Both a and d are correct.
7. Consider the formula studied in class for predicting the next length of a process’s CPU burst: 

**Let α be 0.4**. If the previous prediction (estimate) is **4** and the actual length of the last CPU burst is **8**, the predicted value of the next CPU burst length according to the above formula is:

1. 2.4
2. 3.2
3. 4.0
4. 5.6
5. 6.4
6. 8.0