

Import Settings:  
Base Settings: Brownstone Default  
Highest Answer Letter: D  
Multiple Keywords in Same Paragraph: No

Chapter: Chapter 9

Multiple Choice

1. Which of the following is a benefit of allowing a program that is only partially in memory to execute?
- A) Programs can be written to use more memory than is available in physical memory.
  - B) CPU utilization and throughput is increased.
  - C) Less I/O is needed to load or swap each user program into memory.
  - D) All of the above

Ans: D  
Feedback: 9.1  
Difficulty: Easy

2. In systems that support virtual memory, \_\_\_\_.
- A) virtual memory is separated from logical memory.
  - B) virtual memory is separated from physical memory.
  - C) physical memory is separated from secondary storage.
  - D) physical memory is separated from logical memory.

Ans: D  
Feedback: 9.1  
Difficulty: Medium

3. The `vfork()` system call in UNIX \_\_\_\_.
- A) allows the child process to use the address space of the parent
  - B) uses copy-on-write with the `fork()` call
  - C) is not intended to be used when the child process calls `exec()` immediately after creation
  - D) duplicates all pages that are modified by the child process

Ans: A  
Feedback: 9.1  
Difficulty: Easy

4. Suppose we have the following page accesses: 1 2 3 4 2 3 4 1 2 1 1 3 1 4 and that there are three frames within our system. Using the FIFO replacement algorithm, what is the number of page faults for the given reference string?

- A) 14
- B) 8
- C) 13
- D) 10

Ans: B  
Feedback: 9.4.2  
Difficulty: Medium

5. Suppose we have the following page accesses: 1 2 3 4 2 3 4 1 2 1 1 3 1 4 and that there are three frames within our system. Using the FIFO replacement algorithm, what will be the final configuration of the three frames following the execution of the given reference string?

- A) 4, 1, 3
- B) 3, 1, 4
- C) 4, 2, 3
- D) 3, 4, 2

Ans: D  
Feedback: 9.4.2  
Difficulty: Medium

6. Suppose we have the following page accesses: 1 2 3 4 2 3 4 1 2 1 1 3 1 4 and that there are three frames within our system. Using the LRU replacement algorithm, what is the number of page faults for the given reference string?

- A) 14
- B) 13
- C) 8
- D) 10

Ans: C  
Feedback: 9.4.4  
Difficulty: Medium

7. Given the reference string of page accesses: 1 2 3 4 2 3 4 1 2 1 1 3 1 4 and a system with three page frames, what is the final configuration of the three frames after the LRU algorithm is applied?

- A) 1, 3, 4
- B) 3, 1, 4
- C) 4, 1, 2
- D) 1, 2, 3

Ans: B

Feedback: 9.4.4

Difficulty: Medium

8. Belady's anomaly states that \_\_\_\_.

- A) giving more memory to a process will improve its performance
- B) as the number of allocated frames increases, the page-fault rate may decrease for all page replacement algorithms
- C) for some page replacement algorithms, the page-fault rate may decrease as the number of allocated frames increases
- D) for some page replacement algorithms, the page-fault rate may increase as the number of allocated frames increases

Ans: D

Feedback: 9.4.2

Difficulty: Medium

9. Optimal page replacement \_\_\_\_.

- A) is the page-replacement algorithm most often implemented
- B) is used mostly for comparison with other page-replacement schemes
- C) can suffer from Belady's anomaly
- D) requires that the system keep track of previously used pages

Ans: B

Feedback: 9.4.3

Difficulty: Medium

10. In the enhanced second chance algorithm, which of the following ordered pairs represents a page that would be the best choice for replacement?

- A) (0,0)
- B) (0,1)
- C) (1,0)
- D) (1,1)

Ans: A

Feedback: 9.4.5.3

Difficulty: Medium

11. The \_\_\_\_\_ allocation algorithm allocates available memory to each process according to its size.

- A) equal
- B) global
- C) proportional
- D) slab

Ans: C

Feedback: 9.5.2

Difficulty: Easy

12. The \_\_\_\_\_ is the number of entries in the TLB multiplied by the page size.

- A) TLB cache
- B) page resolution
- C) TLB reach
- D) hit ratio

Ans: C

Feedback: 9.9.3

Difficulty: Easy

13. \_\_\_\_\_ allows the parent and child processes to initially share the same pages, but when either process modifies a page, a copy of the shared page is created.

- A) copy-on-write
- B) zero-fill-on-demand
- C) memory-mapped
- D) virtual memory fork

Ans: A

Feedback: 9.3

Difficulty: Medium

14. \_\_\_\_\_ is the algorithm implemented on most systems.

- A) FIFO
- B) Least frequently used
- C) Most frequently used
- D) LRU

Ans: D

Feedback: 9.4

Difficulty: Medium

15. \_\_\_\_\_ occurs when a process spends more time paging than executing.

- A) Thrashing
- B) Memory-mapping
- C) Demand paging
- D) Swapping

Ans: A

Feedback: 9.6

Difficulty: Easy

16. Windows uses a local page replacement policy \_\_\_\_\_.

- A) when a process exceeds its working set minimum
- B) when a process exceeds its working set maximum
- C) when the system undergoes automatic working set trimming
- D) under all circumstances

Ans: B

Feedback: 9.10.1

Difficulty: Easy

17. Which of the following statements is false with regard to Solaris memory management?

- A) The speed at which pages are examined (the scanrate) is constant.
- B) The pageout process only runs if the number of free pages is less than lotsfree.
- C) An LRU approximation algorithm is employed.
- D) Pages selected for replacement may be reclaimed before being placed on the free list.

Ans: A

Feedback: 9.10.2  
Difficulty: Medium

18. What size segment will be allocated for a 39 KB request on a system using the Buddy system for kernel memory allocation?

- A) 39 KB
- B) 42 KB
- C) 64 KB
- D) None of the above

Ans: C

Feedback: 9.8.1  
Difficulty: Easy

19. Which of the following statements is false with regard to allocating kernel memory?

- A) Slab allocation does not suffer from fragmentation.
- B) Adjacent segments can be combined into one larger segment with the buddy system.
- C) Because the kernel requests memory of varying sizes, some of which may be quite small, the system does not have to be concerned about wasting memory.
- D) The slab allocator allows memory requests to be satisfied very quickly.

Ans: C

Feedback: 9.8  
Difficulty: Medium

20. The \_\_\_\_\_ is an approximation of a program's locality.

- A) locality model
- B) working set
- C) page fault frequency
- D) page replacement algorithm

Ans: B

Feedback: 9.6.2  
Difficulty: Medium

21. \_\_\_\_\_ allows a portion of a virtual address space to be logically associated with a file.

- A) Memory-mapping
- B) Shared memory

- C) Slab allocation
- D) Locality of reference

Ans: A

Feedback: 9.7

Difficulty: Medium

22. Systems in which memory access times vary significantly are known as \_\_\_\_\_.

- A) memory-mapped I/O
- B) demand-paged memory
- C) non-uniform memory access
- D) copy-on-write memory

Ans: C

Feedback: 9.5.4

Difficulty: Medium

23. Which of the following is considered a benefit when using the slab allocator?

- A) Memory is allocated using a simple power-of-2 allocator.
- B) It allows kernel code and data to be efficiently paged.
- C) It allows larger segments to be combined using coalescing.
- D) There is no memory fragmentation.

Ans: D

Feedback: 9.8.2

Difficulty: Medium

Essay

24. Explain the distinction between a demand-paging system and a paging system with swapping.

Ans: A demand-paging system is similar to a paging system with swapping where processes reside in secondary memory. With demand paging, when a process is executed, it is swapped into memory. Rather than swapping the entire process into memory, however, a lazy swapper is used. A lazy swapper never swaps a page into memory unless that page will be needed. Thus, a paging system with swapping manipulates entire processes, whereas a demand pager is concerned with the individual pages of a process.

Feedback: 9.2  
Difficulty: Difficult

25. Explain the sequence of events that happens when a page-fault occurs.

Ans: When the operating system cannot load the desired page into memory, a page-fault occurs. First, the memory reference is checked for validity. In the case of an invalid request, the program will be terminated. If the request was valid, a free frame is located. A disk operation is then scheduled to read the page into the frame just found, update the page table, restart the instruction that was interrupted because of the page fault, and use the page accordingly.

Feedback: 9.2  
Difficulty: Medium

26. How is the effective access time computed for a demand-paged memory system?

Ans: In order to compute the effective access time, it is necessary to know the average memory access time of the system, the probability of a page fault, and the time necessary to service a page fault. The effective access time can then be computed using the formula:

$$\text{effective access time} = (1 - \text{probability of page fault}) * \text{memory access time} + \text{probability of page fault} * \text{page fault time.}$$

Feedback: 9.2.2  
Difficulty: Medium

27. How does the second-chance algorithm for page replacement differ from the FIFO page replacement algorithm?

Ans: The second-chance algorithm is based on the FIFO replacement algorithm and even degenerates to FIFO in its worst-case scenario. In the second-chance algorithm, a FIFO replacement is implemented along with a reference bit. If the reference bit is set, then it is cleared, the page's arrival time is set to the current time, and the program moves along in a similar fashion through the pages until a page with a cleared reference bit is found and subsequently replaced.

Feedback: 9.4  
Difficulty: Medium

28. Explain the concept behind prepaging.

Ans: Paging schemes, such as pure demand paging, result in large amounts of initial page faults

as the process is started. Prepaging is an attempt to prevent this high level of initial paging by bringing into memory, at one time, all of the pages that will be needed by the process.

Feedback: 9.9.1

Difficulty: Medium

29. Why doesn't a local replacement algorithm solve the problem of thrashing entirely?

Ans: With local replacement, if one process starts thrashing, it cannot steal frames from another process and cause the latter to thrash as well. However, if processes are thrashing, they will be in the queue for the paging device most of the time. The average service time for a page fault will increase because of the longer average queue for the paging device. Thus, the effective access time will increase, even for a process that is not thrashing.

Feedback: 9.6

Difficulty: Medium

30. Explain the difference between programmed I/O (PIO) and interrupt driven I/O.

Ans: To send out a long string of bytes through a memory-mapped serial port, the CPU writes one data byte to the data register to signal that it is ready for the next byte. If the CPU uses polling to watch the control bit, constantly looping to see whether the device is ready, this method of operation is called programmer I/O. If the CPU does not poll the control bit, but instead receives an interrupt when the device is ready for the next byte, the data transfer is said to be interrupt driven.

Feedback: 9.7.3

Difficulty: Medium

31. What are the benefits of using slab allocation to allocate kernel memory?

Ans: The slab allocator provides two main benefits. First, no memory is wasted due to fragmentation. When the kernel requests memory for an object, the slab allocator returns the exact amount of memory required to represent the object. Second, memory requests can be satisfied quickly. Objects are created in advance and can be quickly allocated. Also, released objects are returned to the cache and marked as free, thus making them immediately available for subsequent requests.

Feedback: 9.8.2

Difficulty: Medium

32. How are lock bits useful in I/O requests?

Ans: A lock bit is associated with every frame. If a frame is locked, it cannot be selected for replacement. To write a block on tape, we lock into memory the pages containing the block. The system then continues as usual with other processes if the I/O request is in a queue for that I/O device. This avoids the replacement of the pages for other processes and the possible unavailability of those pages when the I/O request advances to the head of the device queue. When the I/O is complete, the pages are unlocked.

Feedback: 9.9.6

Difficulty: Medium

33. Explain how copy-on-write operates.

Ans: Copy-on-write (COW) initially allows a parent and child process to share the same pages. As long as either process is only reading—and not modifying—the shared pages, both processes can share the same pages, thus increasing system efficiency. However, as soon as either process modifies a shared page, a copy of that shared page is created, thus providing each process with its own private page. For example, assume an integer  $X$  whose value is 5 is in a shared page marked as COW. The parent process then proceeds to modify  $X$ , changing its value to 10. Since this page is marked as COW, a copy of the page is created for the parent process, which changes the value of  $X$  to 10. The value of  $X$  remains at 5 for the child process.

Feedback: 9.3

Difficulty: Medium

34. Explain the distinction between global allocation versus local allocation.

Ans: When a process incurs a page fault, it must be allocated a new frame for bringing the faulting page into memory. The two general strategies for allocating a new frame are global and local allocation policies. In a global allocation scheme, a frame is allocated from any process in the system. Thus, if process A incurs a page fault, it may be allocated a page from process B. The page that is selected from process B may be based upon any of the page replacement algorithms such as LRU. Alternatively, a local allocation policy dictates that when a process incurs a page fault, it must select one of its own pages for replacement when allocating a new page.

Feedback: 9.5.3

Difficulty: Medium

35. Discuss two strategies for increasing TLB reach.

Ans: TLB reach refers to the amount of memory accessible from the TLB and is the page size multiplied by the number of entries in the TLB. Two possible approaches for increasing TLB

reach are (1) increasing the number of entries in the TLB, and (2) increasing the page size. Increasing the number of entries in the TLB is a costly strategy as the TLB consists of associative memory, which is both costly and power hungry. For example, by doubling the number of entries in the TLB, the TLB reach is doubled. However, increasing the page size (or providing multiple page sizes) allows system designers to maintain the size of the TLB, and yet significantly increase the TLB reach. For this reason, recent trends have moved towards increasing page sizes for increasing TLB reach.

Feedback: 9.9.3

Difficulty: Medium

36. What is the benefit of using sparse addresses in virtual memory?

Ans: Virtual address spaces that include holes between the heap and stack are known as sparse address spaces. Using a sparse address space is beneficial because the holes can be filled as the stack or heap segments grow, or when we wish to dynamically link libraries (or possibly other shared objects) during program execution.

Feedback: 9.1

Difficulty: Medium

37. Explain the usefulness of a modify bit.

Ans: A modify bit is associated with each page frame. If a frame is modified (i.e. written), the modify bit is then set. The modify bit is useful when a page is selected for replacement. If the bit is not set (the page was not modified), the page does not need to be written to disk. If the modify bit is set, the page needs to be written to disk when selected for replacement.

Feedback: 9.4.1

Difficulty: Medium

True/False

38. In general, virtual memory decreases the degree of multiprogramming in a system.

Ans: False

Feedback: 9.1

Difficulty: Easy

39. Stack algorithms can never exhibit Belady's anomaly.

Ans: True  
Feedback: 9.4  
Difficulty: Medium

40. If the page-fault rate is too high, the process may have too many frames.

Ans: False  
Feedback: 9.6  
Difficulty: Medium

41. The buddy system for allocating kernel memory is very likely to cause fragmentation within the allocated segments.

Ans: True  
Feedback: 9.8.1  
Difficulty: Easy

42. On a system with demand-paging, a process will experience a high page fault rate when the process begins execution.

Ans: True  
Feedback: 9.2  
Difficulty: Easy

43. On systems that provide it, `vfork()` should always be used instead of `fork()`.

Ans: False  
Feedback: 9.3  
Difficulty: Medium

44. Only a fraction of a process's working set needs to be stored in the TLB.

Ans: False  
Feedback: 9.9.3

Difficulty: Medium

45. Solaris uses both a local and global page replacement policy.

Ans: False

Feedback: 9.10.2

Difficulty: Easy

46. Windows uses both a local and global page replacement policy.

Ans: False

Feedback: 9.10.3

Difficulty: Easy

47. A page fault must be preceded by a TLB miss.

Ans: True

Feedback: 9.2.1

Difficulty: Medium

48. Non-uniform memory access has little effect on the performance of a virtual memory system.

Ans: False

Feedback: 9.5.4

Difficulty: Medium

49. In Linux, a slab may only be either full or empty.

Ans: False

Feedback: 9.8.2

Difficulty: Medium

