WAYGROUND formerly Quizizz **Worksheets**

Name

OS UNIT-4

Class

Total questions: 99

Worksheet time: 8hrs 15mins

Date

1. What is the main purpose of memory management in an operating system?

   a) To increase the size of the CPU

   b) To manage how processes share and use memory efficiently

   c) To reduce the number of processes in the system

   d) To slow down the computer's response to users

2. Which of the following is NOT a strategy for allocating memory contiguously?

   a) First-fit

   b) Best-fit

   c) Worst-fit

   d) Round-robin

3. What is the role of the memory management unit (MMU)?

   a) To execute user programs

   b) To translate logical addresses to physical addresses

   c) To schedule CPU tasks

   d) To manage disk storage

4. Why do most memory-management algorithms require hardware support?

   a) Because software cannot manage memory

   b) Because hardware is faster at running user applications

   c) Because closely integrated hardware and operating-system memory management is needed

   d) Because hardware is cheaper than software

5. The selection of a memory-management method depends on the hardware design of the system because:

   a) Different hardware designs require different memory-management methods for optimal performance and integration.

   b) Hardware design does not affect memory management.

   c) All systems use the same memory-management method.

   d) Hardware design only affects CPU speed, not memory management.

6. Which of the following is the only general-purpose storage that the CPU can access directly?

a) A) Main memory and registers
b) B) Disk storage and registers
c) C) Cache and disk storage
d) D) Registers and hard drive

7. What is the main reason for a processor stall during memory access?

a) A) The CPU is overheating
b) B) The data required to complete the instruction is not available in main memory
c) C) The CPU clock is too slow
d) D) The operating system is updating

8. Which hardware component is typically added to the CPU chip to speed up memory access?

a) A) Hard drive
b) B) Cache
c) C) Register
d) D) Disk controller

9. Why must hardware protect the operating system from user processes?

a) A) To increase CPU speed
b) B) To ensure correct operation and prevent unauthorized access
c) C) To reduce memory usage
d) D) To allow direct disk access

10. A multithreaded core can switch from the stalled hardware thread to another hardware thread during which event?

a) A) CPU overclocking
b) B) Memory stall
c) C) Register overflow
d) D) Disk failure

11. What is the primary purpose of having separate per-process memory spaces in an operating system?

a) To protect processes from each other and allow concurrent execution
b) To increase the speed of the CPU
c) To reduce the size of the operating system
d) To allow processes to share memory freely

12. Which two registers are used to define the range of legal addresses a process may access?

a) Base and limit registers
b) Stack and heap registers
c) Program counter and instruction register
d) Accumulator and index register

13. If a base register holds 300040 and the limit register is 120900, what is the range of legal addresses the program can access?

a) 300040 through 420939 (inclusive)
b) 0 through 120900 (inclusive)
c) 120900 through 300040 (inclusive)
d) 300040 through 120900 (inclusive)

14. How does the CPU hardware protect memory spaces when a program executes in user mode?

a) By comparing every address generated with the base and limit registers
b) By encrypting all memory addresses
c) By allowing direct access to all memory
d) By disabling the operating system

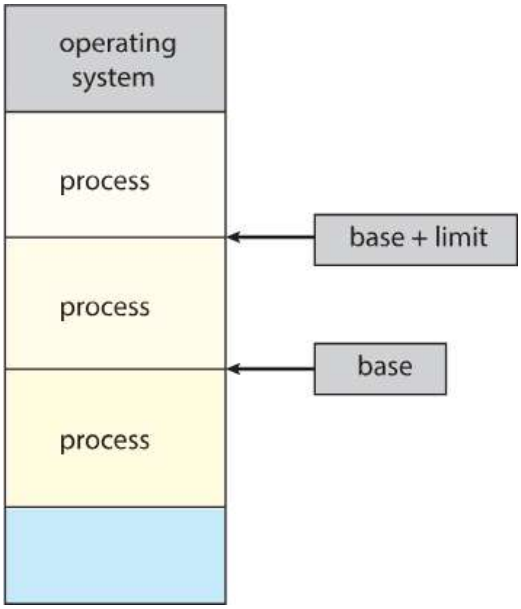
15. Why can only the operating system load the base and limit registers?

a) Because it requires a special privileged instruction executed only in kernel mode
b) Because user programs are not allowed to access memory
c) Because the registers are read-only
d) Because the CPU does not support user access

16. What is a consequence if a user program tries to access memory outside its legal range?

a) A trap to the operating system occurs, treated as a fatal error
b) The program continues without interruption
c) The memory is automatically expanded
d) The operating system ignores the attempt

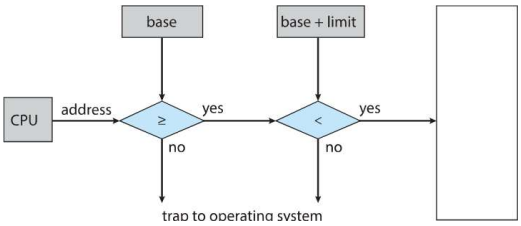
17.



What does the diagram illustrate regarding memory management?

a) How base and limit registers define a logical address space for processes
b) How processes communicate with each other
c) How the CPU executes instructions
d) How files are stored on disk

18.



Which hardware mechanism is used to protect memory addresses in a system, as shown in Figure?

a) Base and limit registers
b) Stack pointer and program counter
c) Cache and TLB
d) Segment and offset registers

19. What is the main purpose of address binding in an operating system?

a) To convert symbolic addresses to physical addresses
b) To increase the speed of the CPU
c) To reduce the size of executable files
d) To encrypt user data

20. At which stage can the binding of instructions and data to memory addresses NOT be done, according to the text?

a) Compile time
b) Link time
c) Run time
d) Post-processing time

21. Why might it be necessary to recompile code if the starting location of a process in memory changes after compile time?

a) Because absolute code is generated based on a fixed starting location
b) Because the CPU speed changes
c) Because the operating system updates automatically
d) Because the user changes the source code

22. Which of the following best describes a symbolic address in a program?

a) An address that refers to a variable or label in the source code
b) An address that is always zero
c) An address that is generated by the operating system
d) An address that is used only by the hardware

23. Which of the following best describes "load time" in the context of program execution?

a) The time when the compiler generates the source code.
b) The time when the final binding of a program to memory is delayed until the program is loaded into memory.
c) The time when the program is executed by the CPU.
d) The time when the linker combines object files.

24. What must the compiler generate if it is not known at compile time where the process will reside in memory?

a) Static code
b) Relocatable code
c) Executable code
d) Linked code

25. At which stage must binding be delayed if a process can be moved during its execution from one memory segment to another?

a) Compile time
b) Load time
c) Execution time (run time)
d) Link time

26. Which of the following is commonly referred to as a logical address?

a) An address generated by the linker
b) An address generated by the CPU
c) An address seen by the memory unit
d) An address generated by the loader

27. Why is special hardware required for binding at execution time?

a) To generate source code
b) To allow the process to move between memory segments during execution
c) To link object files
d) To compile the program

28. What is the function of the Memory Management Unit (MMU) in a computer system?

a) It converts logical addresses to physical addresses.
b) It stores data permanently.
c) It manages CPU speed.
d) It controls input/output devices.

29. Which term is used interchangeably with "logical address" in the context of memory management?

a) Virtual address
b) Physical address
c) Relocation register
d) Memory-address register

30. What is the set of all logical addresses generated by a program called?

a) Logical address space
b) Physical address space
c) Memory register
d) Relocation register

31. What is the set of all physical addresses corresponding to logical addresses called?

a) Physical address space
b) Logical address space
c) Virtual address space
d) Memory management unit

32. What is the value in the relocation register used for?

a) It is added to every address generated by a user process at the time the address is sent to memory.
b) It stores the logical address.
c) It manages CPU instructions.
d) It controls the speed of memory access.

33. Why does the user program never access the real physical addresses directly?

a) Because the mapping hardware converts logical addresses into physical addresses.
b) Because physical addresses are not used in memory.
c) Because logical addresses are faster.
d) Because the CPU only understands physical addresses.

34. If the base in the relocation register is set at 14000, what would be the physical address for a logical address 346?

a) 14346
b) 346
c) 14000
d) 14600

35. What must happen before logical addresses generated by a user program can be used?

a) They must be mapped to physical addresses.
b) They must be stored in the CPU.
c) They must be converted to virtual addresses.
d) They must be deleted.

36. What is the main advantage of dynamic loading in memory management?

a) The entire program must be loaded into memory at once.
b) Routines are loaded only when they are called.
c) All routines are kept in physical memory permanently.
d) The program size must be limited to physical memory size.

37. Which of the following best describes dynamically linked libraries (DLLs)?

a) Libraries that are linked to user programs only at compile time.
b) Libraries that are linked to user programs when the programs are run.
c) Libraries that are never linked to user programs.
d) Libraries that are linked to user programs only during system boot.

38. Why is dynamic loading particularly useful for handling error routines?

a) Error routines must always be loaded in memory.
b) Error routines are frequently used and require fast access.
c) Error routines occur infrequently and only need to be loaded when required.
d) Error routines are part of the main program and cannot be loaded separately.

39. A programmer wants to reduce the memory usage of a program by loading routines only when needed. Which technique should they use?

a) Static linking
b) Dynamic loading
c) Physical memory allocation
d) Permanent memory mapping

40. Which responsibility does dynamic loading place on the user?

a) Users must design their programs to take advantage of dynamic loading.
b) Users must always load the entire program into memory.
c) Users must avoid using any library routines.
d) Users must rely on the operating system to load all routines automatically.

41. What is the difference between static linking and dynamic linking?

a) Static linking links libraries at runtime, dynamic linking links at compile time.
b) Static linking links libraries at compile time, dynamic linking links at runtime.
c) Both static and dynamic linking link libraries at runtime.
d) Both static and dynamic linking link libraries at compile time.

42. What is the main advantage of using DLLs (Dynamic Link Libraries) in operating systems like Windows and Linux?

a) They allow only one instance of the DLL in main memory to be shared among multiple processes.
b) They increase the size of the executable image.
c) They prevent the use of system libraries.
d) They require each program to have its own copy of the library.

43. Which of the following best describes dynamic linking?

a) Linking is postponed until execution time, allowing libraries to be loaded as needed.
b) All libraries are loaded at compile time.
c) Libraries are never updated after the program is compiled.
d) Programs must always be recompiled to use new library versions.

44. What is a potential disadvantage of not using dynamic linking in programs?

a) Programs would need to be relinked to gain access to new library versions.
b) Programs would always use the latest library version automatically.
c) Programs would never need to be updated.
d) Programs would be protected from incompatible library changes.

45. Why do dynamically linked libraries generally require help from the operating system?

a) To check if the needed routine is in another process's memory space or to allow multiple processes to access the same memory addresses.

b) To prevent any program from accessing libraries.

c) To ensure that only one program can use a library at a time.

d) To automatically update all programs to the latest library version.

46. According to the text, what is the usual way main memory is divided for allocation?

a) Into two partitions: one for the operating system and one for user processes.

b) Into three partitions: one for the operating system, one for user processes, and one for libraries.

c) Into equal partitions for each process.

d) Into partitions based on the size of the executable image.

47. Which memory region do most operating systems, including Linux and Windows, place the operating system in?

a) High memory

b) Low memory

c) Middle memory

d) Random memory

48. What is contiguous memory allocation?

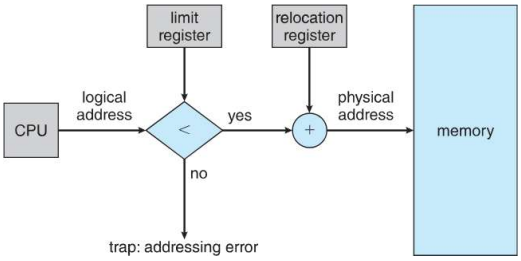
a) Each process is contained in a single section of memory that is contiguous to the section containing the next process.

b) Each process is scattered randomly throughout memory.

c) Each process shares memory with other processes.

d) Each process is stored in external storage.

49.



What is the purpose of the relocation register in memory protection?

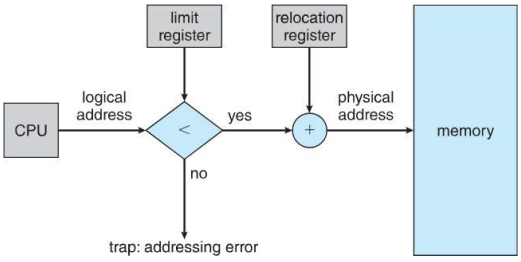
a) It contains the value of the smallest physical address.

b) It contains the largest logical address.

c) It stores the CPU instructions.

d) It manages device drivers.

50.



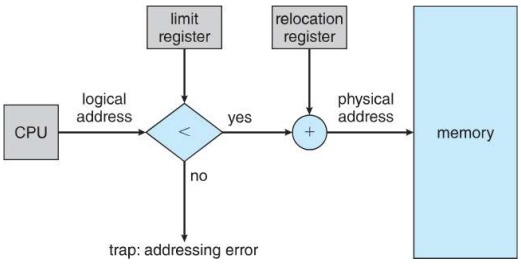
How does the MMU map the logical address to the physical address?

a) By adding the value in the relocation register to the logical address.
b) By subtracting the value in the limit register from the logical address.
c) By multiplying the logical address by the relocation register.
d) By dividing the logical address by the limit register.

51. Why is the relocation-register scheme considered flexible for operating systems?

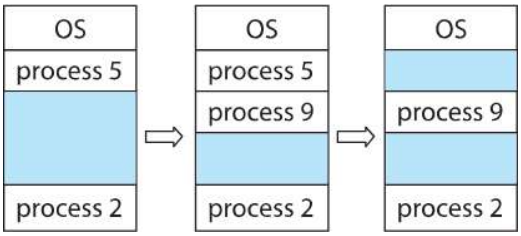
a) It allows the operating system's size to change dynamically.
b) It prevents all errors in memory allocation.
c) It increases the speed of the CPU.
d) It eliminates the need for device drivers.

52.



What happens if a logical address generated by the CPU does not fall within the range specified by the limit register?

a) A trap for addressing error occurs.
b) The address is automatically corrected.
c) The process is terminated immediately.
d) The address is sent to external storage.

53. 

Which of the following best describes the variable-partition scheme in memory allocation?

a) A. Each partition may contain multiple processes.
b) B. Each partition may contain exactly one process.
c) C. Each partition is fixed in size.
d) D. Each partition is reserved for the operating system only.

54. What happens when a process terminates in a variable-partition memory allocation scheme?

a) A. Its memory is permanently lost.
b) B. Its memory is released and can be allocated to another process.
c) C. Its memory is transferred to the operating system.
d) D. Its memory is merged with the CPU cache.

55. Which strategies are most commonly used to select a free hole from the set of available holes in dynamic storage-allocation?

a) A. Least-fit, random-fit, and average-fit
b) B. First-fit, best-fit, and worst-fit
c) C. Quick-fit, slow-fit, and medium-fit
d) D. Smallest-fit, largest-fit, and middle-fit

56. If a hole in memory is too large for a process, what is the typical procedure followed?

a) A. The process is rejected.
b) B. The hole is split into two parts.
c) C. The hole is merged with another hole.
d) D. The hole is ignored.

57. Explain how the operating system manages memory allocation when there isn't sufficient memory to satisfy the demands of an arriving process.

a) A. It immediately allocates memory from the CPU cache.
b) B. It rejects the process or places it in a wait queue until memory is available.
c) C. It merges all holes to create a larger block.
d) D. It compresses the memory to fit the process.

58. What is the dynamic storage-allocation problem concerned with?

a) A. How to allocate memory to the operating system only.
b) B. How to satisfy a request of size n from a list of free holes.
c) C. How to permanently store processes in memory.
d) D. How to allocate memory randomly.

59. Describe the process that occurs when a new hole is adjacent to other holes in memory.

a) A. The holes are ignored.
b) B. The holes are merged to form one larger hole.
c) C. The holes are allocated to the operating system.
d) D. The holes are split into smaller holes.

60. Which memory allocation strategy allocates the first hole that is big enough for the process?

a) First fit
b) Best fit
c) Worst fit
d) Next fit

61. Which memory allocation strategy produces the smallest leftover hole?

a) Best fit
b) First fit
c) Worst fit
d) Next fit

62. Which memory allocation strategy allocates the largest hole available?

a) Worst fit
b) Best fit
c) First fit
d) Quick fit

63. Simulations have shown that which two strategies are better than worst fit in terms of decreasing time and storage utilization?

a) First fit and best fit
b) Best fit and worst fit
c) First fit and next fit
d) Worst fit and next fit

64. What is external fragmentation?

a) When total memory space is enough but available spaces are not contiguous
b) When memory is allocated in fixed-sized blocks
c) When memory is allocated in variable-sized blocks
d) When memory is not allocated at all

65. According to statistical analysis, what fraction of memory may be unusable due to fragmentation in first fit allocation?

a) One-third
b) One-fourth
c) One-half
d) One-fifth

66. What is internal fragmentation?

a) Unused memory that is internal to a partition
b) Unused memory between partitions
c) Unused memory at the end of the memory
d) Unused memory in the cache

67. Why might external fragmentation be a problem regardless of the allocation algorithm used?

a) Because storage is fragmented into many small holes
b) Because memory is always allocated in large blocks
c) Because processes never release memory
d) Because memory is never reused

68. What is the main goal of compaction in memory management?

a) To shuffle memory contents and place all free memory together in one large block.
b) To increase the speed of the CPU.
c) To reduce the size of the operating system.
d) To allocate more memory to each process.

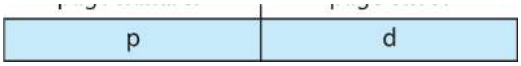
69. Which memory management technique allows a process's physical address space to be noncontiguous?

a) Compaction
b) Paging
c) Segmentation
d) Fragmentation

70. What is the basic method for implementing paging?

a) Breaking physical memory into fixed-sized blocks called frames and logical memory into blocks of the same size called pages.
b) Allocating all memory to a single process.
c) Dividing memory based on process priority.
d) Using only dynamic relocation for memory allocation.

71.



What are the two parts into which every address generated by the CPU is divided?

a) Page number and page offset
b) Frame number and segment offset
c) Block number and block offset
d) Memory address and logical address

72. Why is compaction not always possible in memory management?

a) Because relocation is static and done at assembly or load time.
b) Because memory is always contiguous.
c) Because paging is always used instead.
d) Because the CPU cannot generate addresses.

73. How does paging help avoid external fragmentation?

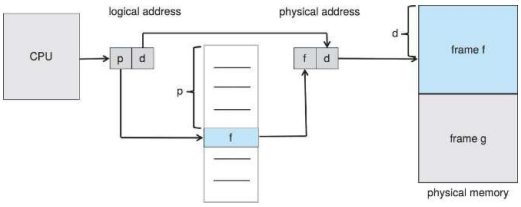
a) By allowing the physical address space to be noncontiguous.
b) By increasing the size of memory.
c) By using only dynamic relocation.
d) By allocating memory only to the operating system.

74. What is the backing store divided into for paging?

a) Fixed-sized blocks that are the same size as memory frames or clusters of multiple frames.
b) Variable-sized blocks based on process size.
c) Segments based on logical address.
d) Pages based on CPU speed.

75. What is a major advantage of paging in modern operating systems?

a) It avoids external fragmentation and the need for compaction.
b) It increases fragmentation.
c) It requires more memory for each process.
d) It only works for mobile devices.

76. 

Which component uses the page number as an index to translate a logical address to a physical address?

a) Memory Management Unit (MMU)
b) Central Processing Unit (CPU)
c) Arithmetic Logic Unit (ALU)
d) Cache Controller

77. What is the first step taken by the MMU to translate a logical address generated by the CPU to a physical address?

a) Extract the corresponding frame number from the page table
b) Replace the page number in the logical address with the frame number
c) Extract the page number and use it as an index into the page table
d) Add the offset to the frame number

78. Which part of the logical address does NOT change during the translation to a physical address?

a) Page number
b) Frame number
c) Offset
d) Base address

79. If the size of the logical address space is 2^m and the page size is 2^n bytes, how many bits of the logical address designate the page offset?

a) m bits
b) n bits
c) m-n bits
d) 2n bits

80. Why is the page size typically chosen as a power of 2 in computer architecture?

a) To maximize memory usage
b) To simplify the translation of logical addresses into page numbers and offsets
c) To increase the speed of the CPU
d) To reduce the size of the page table

81. Given a logical address divided into a page number and a page offset, what do the high-order bits represent?

a) The frame number
b) The page offset
c) The page number
d) The physical address

82. Which of the following best describes the purpose of a page table in a paging memory management scheme?

a) It maps logical addresses to physical addresses.
b) It stores the data of each page.
c) It manages the CPU scheduling.
d) It controls input/output devices.

83. Given a page size of 4 bytes and a physical memory of 32 bytes (8 pages), what is the physical address for logical address 0 (page 0, offset 0) if page 0 is in frame 5?

a) 20
b) 0
c) 5
d) 32

84. If logical address 3 corresponds to page 0, offset 3, and page 0 is in frame 5, what is the physical address?

a) 23
b) 3
c) 15
d) 8

85. What type of fragmentation is eliminated by using a paging scheme?

a) External fragmentation
b) Internal fragmentation
c) Both internal and external fragmentation
d) No fragmentation

86. A process of 72,766 bytes with a page size of 2,048 bytes will be allocated how many frames, and what is the resulting internal fragmentation?

a) 36 frames, 962 bytes
b) 35 frames, 1,086 bytes
c) 36 frames, 1,086 bytes
d) 35 frames, 962 bytes

87. Why does internal fragmentation occur in a paging scheme?

a) Because frames are allocated as units and the last frame may not be completely full
b) Because pages are of different sizes
c) Because logical addresses are not mapped
d) Because physical memory is not divided into frames

88. (DoK Level 3) Suppose a process requires n pages plus 1 byte. According to the paging scheme, how many frames will be allocated, and what is the implication for internal fragmentation?

a) n + 1 frames; almost an entire frame may be wasted as internal fragmentation
b) n frames; no internal fragmentation
c) n frames; only 1 byte wasted
d) n + 1 frames; no internal fragmentation

89. What is the typical size of a page in modern systems as mentioned in the text?

a) 4 KB or 8 KB
b) 16 KB or 32 KB
c) 1 KB or 2 KB
d) 64 KB or 128 KB

90. What is the main advantage of using larger page sizes in a paging system?

a) Reduced overhead in each page-table entry
b) Increased internal fragmentation
c) Decreased efficiency of disk I/O
d) Smaller logical memory size

91. On a 32-bit CPU, how many physical page frames can a 32-bit page-table entry point to?

a) $2^3 2$
b) 2^{16}
c) 2^8
d) 2^4

92. If the frame size is 4 KB, what is the maximum physical memory that a system with 4-byte entries can address?

a) 16 TB
b) 4 GB
c) 1 TB
d) 32 GB

93. Why is the size of physical memory in a paged memory system typically different from the maximum logical size of a process?

a) Because logical memory is mapped to physical memory using page tables
b) Because physical memory is always larger than logical memory
c) Because logical memory is not used in paging systems
d) Because page sizes are always fixed

94. Which command can be used on a Linux system to obtain the page size as a number of bytes?

a) getconf PAGESIZE
b) ls -l
c) cat /proc/meminfo
d) ps aux

95. What must be true for a process that requires n pages to be loaded into memory?

a) At least n frames must be available in memory.
b) At least n+1 frames must be available in memory.
c) At least n-1 frames must be available in memory.
d) Only one frame must be available in memory.

96. What is an important aspect of paging in memory management?

a) It separates the programmer's view of memory from the actual physical memory.
b) It increases the speed of the CPU.
c) It eliminates the need for a page table.
d) It combines all processes into a single memory block.

97. When a process arrives in the system, how is its size expressed for memory allocation?

a) In pages
b) In bytes
c) In kilobytes
d) In megabytes

98. What happens to the frame number when a page of a process is loaded into memory?

a) It is put in the page table for the process.
b) It is deleted from memory.
c) It is stored in the CPU cache.
d) It is sent to the printer.

99. What is the main purpose of the frame table in an operating system?

a) To store user process instructions
b) To keep track of physical page frames and their allocation status
c) To translate logical addresses to physical addresses
d) To manage CPU scheduling

Answer Keys

1. b) To manage how processes share and use memory efficiently
2. d) Round-robin
3. b) To translate logical addresses to physical addresses
4. c) Because closely integrated hardware and operating-system memory management is needed
5. a) Different hardware designs require different memory-management methods for optimal performance and integration.
6. a) A) Main memory and registers
7. b) B) The data required to complete the instruction is not available in main memory
8. b) B) Cache
9. b) B) To ensure correct operation and prevent unauthorized access
10. b) B) Memory stall
11. a) To protect processes from each other and allow concurrent execution
12. a) Base and limit registers
13. a) 300040 through 420939 (inclusive)
14. a) By comparing every address generated with the base and limit registers
15. a) Because it requires a special privileged instruction executed only in kernel mode
16. a) A trap to the operating system occurs, treated as a fatal error
17. a) How base and limit registers define a logical address space for processes
18. a) Base and limit registers
19. a) To convert symbolic addresses to physical addresses
20. d) Post-processing time
21. a) Because absolute code is generated based on a fixed starting location
22. a) An address that refers to a variable or label in the source code
23. b) The time when the final binding of a program to memory is delayed until the program is loaded into memory.
24. b) Relocatable code

25. c) Execution time (run time)
26. b) An address generated by the CPU
27. b) To allow the process to move between memory segments during execution
28. a) It converts logical addresses to physical addresses.
29. a) Virtual address
30. a) Logical address space
31. a) Physical address space
32. a) It is added to every address generated by a user process at the time the address is sent to memory.
33. a) Because the mapping hardware converts logical addresses into physical addresses.
34. a) 14346
35. a) They must be mapped to physical addresses.
36. b) Routines are loaded only when they are called.
37. b) Libraries that are linked to user programs when the programs are run.
38. c) Error routines occur infrequently and only need to be loaded when required.
39. b) Dynamic loading
40. a) Users must design their programs to take advantage of dynamic loading.
41. b) Static linking links libraries at compile time, dynamic linking links at runtime.
42. a) They allow only one instance of the DLL in main memory to be shared among multiple processes.
43. a) Linking is postponed until execution time, allowing libraries to be loaded as needed.
44. a) Programs would need to be relinked to gain access to new library versions.
45. a) To check if the needed routine is in another process's memory space or to allow multiple processes to access the same memory addresses.
46. a) Into two partitions: one for the operating system and one for user processes.
47. a) High memory
48. a) Each process is contained in a single section of memory that is contiguous to the section containing the next process.

49. a) It contains the value of the smallest physical address.

50. a) By adding the value in the relocation register to the logical address.

51. a) It allows the operating system's size to change dynamically.

52. a) A trap for addressing error occurs.

53. b) B. Each partition may contain exactly one process.

54. b) B. Its memory is released and can be allocated to another process.

55. b) B. First-fit, best-fit, and worst-fit

56. b) B. The hole is split into two parts.

57. b) B. It rejects the process or places it in a wait queue until memory is available.

58. b) B. How to satisfy a request of size n from a list of free holes.

59. b) B. The holes are merged to form one larger hole.

60. a) First fit

61. a) Best fit

62. a) Worst fit

63. a) First fit and best fit

64. a) When total memory space is enough but available spaces are not contiguous

65. a) One-third

66. a) Unused memory that is internal to a partition

67. a) Because storage is fragmented into many small holes

68. a) To shuffle memory contents and place all free memory together in one large block.

69. b) Paging

70. a) Breaking physical memory into fixed-sized blocks called frames and logical memory into blocks of the same size called pages.

71. a) Page number and page offset

72. a) Because relocation is static and done at assembly or load time.

73. a) By allowing the physical address space to be noncontiguous.

74. a) Fixed-sized blocks that are the same size as memory frames or clusters of multiple frames.

75. a) It avoids external fragmentation and the need for compaction.

76. a) Memory Management Unit (MMU)

77. c) Extract the page number and use it as an

78. c) Offset

index into the page table

79. b) n bits
80. b) To simplify the translation of logical addresses into page numbers and offsets
81. c) The page number
82. a) It maps logical addresses to physical addresses.
83. a) 20
84. a) 23
85. a) External fragmentation
86. a) 36 frames, 962 bytes
87. a) Because frames are allocated as units and the last frame may not be completely full
88. a) n + 1 frames; almost an entire frame may be wasted as internal fragmentation
89. a) 4 KB or 8 KB
90. a) Reduced overhead in each page-table entry
91. a) $2^3 2$
92. a) 16 TB
93. a) Because logical memory is mapped to physical memory using page tables
94. a) getconf PAGESIZE
95. a) At least n frames must be available in memory.
96. a) It separates the programmer's view of memory from the actual physical memory.
97. a) In pages
98. a) It is put in the page table for the process.
99. b) To keep track of physical page frames and their allocation status