# If a magnetic disc drive has 100 cylinders, each containing 10 tracks of 10 sectors, and each sector cancontain 128 bytes, what is the maximum capacity of the disc drive in KB?

A. 160,000 B. 1,280

C. 1,250 D. 1,280,000

The maximum capacity of the disk is 1280000 bytes.

Given data :

1 sector contains 128 bytes.

There are 10 sectors in one track.

One cylinder contains 10 tracks.

There are 100 cylinders.

• The total number of tracks

= Number of cylinders × Tracks in one cylinder

= 100×10

= 1000

• Number of sectors

= Number of tracks × Sectors in one track

= 1000×10

= 10000

• Total amount of data

= Total number of sector × data present in one sector

= 10000×128

= 1280000 bytes

The total capacity of data in terms of bytes is 1280000 bytes

# According to the specifications of a particular hard disk, a seek time takes 0.3 milliseconds betweenadjacent tracks. If the disk has 100 cylinders how long will it take for the head to move from the innermost cylinder to the outermost cylinder.

* 1. 30 milliseconds B. 300 microseconds

C. 3000 microseconds D 0.3 seconds

# A computer that is advertised as having a 96K byte DRAM memory and a 2.1 Gigabyte hard drivehas

* 1. 96K bytes of secondary memory and 2.1 Gigabytes of primary memory
  2. 2.1 Gigabytes of auxiliary memory and 96 K bytes of primary memory
  3. 96K bytes of cache, 2.1 Gigabytes of primary memory
  4. 2.1 Gigabytes of auxiliary memory, 96 K bytes of primary memory and 96 bytes of cache

# The average time required to reach a storage location in memory and obtain its contents is called the

* 1. Seek time B. Turnaround time

C. Access time D. Transfer time

# The amount of time required to read a block of data from a disk into memory is composed of seektime, rotational latency, and transfer time. Rotational latency refers to

* 1. The time it takes for the platter to make a full rotation
  2. The time it takes for the read-write head to move into position over the appropriate track
  3. The time it takes for the platter to rotate the correct sector under the head
  4. None of the others

# The method of placing the heads and the discs in an air tight environment is called as ……..

* 1. RAID Arrays B. ATP tech

C. Winchester technology D. Fleming reduction

**Explanation**: <numeric> The Disks and the heads operate faster due to the absence of the dust particles.

# Which of the following is a component of the disk system?

* 1. Disk B. Disk drive

C. Disk controller D. All of the others

# If the drive has 20 surfaces, how many heads will it have?

A. 1 B. 5

C. 10 D. 20

**Explanation**: <numeric> Each surface will have its own head to perform read/write operation.

# The process divides the disk into sectors and tracks.

* 1. Creation B. Initiation

C. Formatting D. Modification

**Explanation**: <numeric> The formatting process deletes the data present and does the creation ofsectors and tracks.

# The data can be accessed from the disk using ……….

* 1. Surface number B. Sector number

C. Track number D. All of the others

# The is the minimum storage unit of a hard drive.

* 1. Track B. Sector

C. Cluster D. Cylinder

# To distinguish between two sectors we make use of ……….

* 1. Inter sector gap B. Splitting bit

C. Numbering bit D. None of the others

**Explanation**: <numeric> This means that we leave <để lại> a little gap between each sectors todifferentiate between them.

# 13 is used to deal with the difference in the transfer rates between the drive and the bus.

A. Data repeaters B. Enhancers

C. Data buffers D. None of the others

**Explanation**: <numeric> The buffers are added to store the data from the fast device and to send it tothe slower device at its rate.

# What common characteristics are shared by all RAID levels?

* 1. Set of physical disk drives viewed by the operating system as a single logical drive
  2. Data are distributed across the physical drives of an array in a scheme known as striping
  3. Redundant disk capacity is used to store parity information, which guarantees data recoverability incase of a disk failure
  4. All of the others

# The solution to the problem of reliability is the introduction of ……….

* 1. Aging B. Scheduling

C. Redundancy D. Disks

# RAID splits file(s) into many segments, and sends the segments to several disks. Files that have beensegmented in this way are called:

* 1. Striped File B. Striped Data

C. Striped Array D. None of the others

# Which of the following is the RAID level no redundant?

A. 0 B. 1

C. 2 D. 3

# Which of the following is the RAID level refers to memory-style ECC organization?

A. 1 B. 2

C. 3 D. 4

# Which of the following is the RAID level distributes parity and data across all the disks?

A. 3 B. 4

C. 5 D. 6

# RAID level 1+0 is used because RAID level 1 provides ………. whereas RAID level 0 provides ……….

* 1. Performance, Redundancy B. Performance, Reliability C. Redundancy, Performance D. Reliability, Performance

# register keeps tracks of the instructions stored in program stored in memory.

* 1. AR (Address Register) B. XR (Index Register)

C. PC (Program Counter) D. AC (Accumulator)

# An interface that provides a method for transferring binary information between internal storage andexternal devices is called as …

* 1. I/O interface B. Input interface

C. Output interface D. I/O bus

# An interface that provides I/O transfer of data directly to or form the memory unit and peripheral is termed as …

* 1. DDA B. Serial interface

C. BR D. DMA

# External, or peripheral, devices include:

* 1. Human readable B. Machine readable

C. Communication D. All of the others

# With the functions of an I/O module, which of the following statements is false:

* 1. Control and timing
  2. A module only connects to a peripheral device
  3. Exchange information with the processor, with peripherals
  4. Data buffers, error detection

# I/O addressing methods:

* 1. Memory-mapped I/O
  2. Isolated I/O
  3. Both memory-mapped I/O and isolated I/O
  4. None of the others

# In memory-mapped I/O…

* 1. The I/O devices and the memory share the same address space
  2. The I/O devices have a seperate address space
  3. The memory and I/O devices have an associated address space
  4. A part of the memory is specifically set aside for the I/O operation

**Explanation**: It’s the different modes of accessing the i/o devices.

# With isolated I/O, ...

* 1. The I/O devices and the memory share the same address space
  2. The I/O devices have a seperate address space
  3. The memory and I/O devices have an associated address space
  4. A part of the memory is specifically set aside for the I/O operation

**Explanation**: It’s the different modes of accessing the i/o devices.

# There are three methods for performing I/O:

* 1. Interrupt-driven I/O, System-driven I/O, DMA
  2. Interrupt-driven I/O, System-driven I/O, Programmed I/O
  3. Programmed I/O, Interrupt-driven I/O, DMA
  4. Programmed I/O, System-driven I/O, DMA

# The method of accessing the I/O devices by repeatedly checking the status flags is …

* 1. Programmed I/O B. Memory-mapped I/O

C. I/O mapped D. None of the others

**Explanation**: In this method the processor constantly checks the status flags , and when it finds that theflag is set it performs the appropriate operation.

# The method of synchronising the processor with the I/O device in which the device sends a signalwhen it is ready is …

* 1. Exceptions B. Signal handling C. Interrupt-driven I/O D. DMA

**Explanation**: This is a method of accessing the I/O devices which gives the complete power to thedevices, enabling them to intimate the processor when they’re ready for transfer.

# The process where in the processor constantly checks the status flags is called as …

* 1. Polling B. Inspection

C. Reviewing D. Echoing

# With Programmed I/O, which of the following statements is false:

* 1. Use input/output commands in the program to exchange data with the I/O ports
  2. Peripherals are active objects in data exchange
  3. When executing the program, encountering input/output commands, the CPU controls data exchangewith peripherals
  4. Peripherals are passive objects in data exchange

# With Programmed I/O, which of the following statements is true:

* 1. This is the simplest method to exchange data
  2. This is the fastest method to data exchange
  3. Complex circuit design
  4. None of the others

# With Interrupt-driven I/O, which of the following statements is false:

* 1. Peripherals are the active object in data exchange
  2. CPU does not have to wait for the availability of peripherals
  3. CPU have to wait for the availability of I/O module
  4. I/O module interrupt CPU when it is in ready state

# With Interrupt-driven I/O, which of the following statements is true:

* 1. Peripherals are the active object in data exchange
  2. The method is fully processed by hardware
  3. CPU is an active object in data exchange
  4. The method is fully processed by software

# The DMA differs from the interrupt mode by …

* 1. The involvement of the processor for the operation
  2. The method accessing the I/O devices
  3. The amount of data transfer possible
  4. Both the involvement of the processor for the operation and the amount of data transfer possible

**Explanation**: DMA is an approach of performing data transfers in bulk between memory and theexternal device without the intervention of the processor.

# The DMA transfers are performed by a control circuit called as …

* 1. Device interface B. DMA controller

C. Data controller D. Overlooker

**Explanation**: The Controller performs the functions that would normally be carried out by the processor.

# In DMA transfers, the required signals and addresses are given by the …

* 1. Processor B. Device drivers

C. DMA controller D. The program itself

**Explanation**: The DMA controller acts like a processor for DMA transfers and overlooks <bỏ qua> the entire process.

# The technique whereby the DMA controller steals the access cycles of the processor to operate iscalled as …

* 1. Fast conning B. Memory Con

C. Cycle stealing D. Memory stealing

**Explanation**: The controller takes over the processor’s access cycles and performs memory operations.

# The operating system is an example of a computer ……….

* 1. Object B. File system

C. Program D. Desktop

# Which of the following is the primary purpose of an operating system?

* 1. To make the most efficient use of the computer hardware
  2. To allow people to use the computer
  3. To keep systems programmers employed
  4. To make computers easier to use

# The key services provided by an OS:

* 1. Create and execute programs
  2. Control access to I/O devices, files and system resources
  3. Accounting, error detection and response
  4. All of the others

# One of the function of operating system is it serves an interface between user and ……….

* 1. Software B. Hardware

C. Utilities D. Data ware

# In an system the user/programmer interacts directly with the computer, usually through a

**keyboard/display terminal to request the execution of a job or to perform a transaction.**

* 1. Batch B. Multiprogramming

C. Interactive D. None of the others

# Which of the following is NOT a function of operating system?

* 1. Resource Manager B. Storage Manager

C. Process Manager D. Software Manager

# Long-term scheduling is:

* 1. The decision to add which programs to the system for processes
  2. The decision to add to the number of processes that are partially or fully in main memory
  3. The decision as to which available process will be executed by the processor.
  4. The decision as to which process's pending I/O request shall be handled by an available I/O device.

# Medium-term scheduling is:

* 1. The decision as to which process's pending I/O request shall be handled by an available I/O device.
  2. The decision as to which available process will be executed by the processor.
  3. The decision to add to the number of processes that are partially or fully in main memory
  4. The decision to add which programs to the system for processes

# Short-term scheduling is:

* 1. The decision to add which programs to the system for processes
  2. The decision as to which available process will be executed by the processor.
  3. The decision to add to the number of processes that are partially or fully in main memory
  4. The decision as to which process's pending I/O request shall be handled by an available I/O device.

# What is a process?

* 1. A program in execution
  2. A \*.exe file
  3. A executable file stored in external memory
  4. None of the others

# What is the purpose of the process?

A. Multiprocessing B. Multiprogramming C. Multicore D. All of the others

# In the process state transition diagram, which state corresponding to a program is admitted by theLong-term scheduler?

* 1. New B. Ready

C. Running D. Halted

# In the process state transition diagram, will initialize the process, moving it to the ready state.

* 1. Long-term scheduler B. Medium-term scheduler

C. Short-term scheduler D. None of the others

# In the process state transition diagram, the transition from the READY state to the RUNNING state indicates that:

* 1. A process was preempted by another process
  2. A process has blocked for a semaphore or other operation
  3. A process is done waiting for an I/O operation
  4. A process was just created

# The state corresponding to the process has terminated and will be destroyed by the OS is called:

* 1. New B. Ready

C. Running D. Halted

# Copying a process from memory to disk to allow space for other processes is called?

* 1. Page Fault B. Deadlock

C. Demand Paging D. Swapping

# The purpose of swapping is:

* 1. To remove processes not in a ready state
  2. To provide for efficient use of main memory for processes execution
  3. To add processes in a ready state to main memory
  4. None of the others

# Swapping is executed by ..........

* 1. Long-term scheduler B. Medium-term scheduler

C. Short-term scheduler D. None of the others

# If a process may be dynamically assigned to different locations in main memory, what is implicationfor the addressing mechanism?

* 1. The addressing mechanism must keep track of the physical addresses of the process
  2. The addressing mechanism must keep track of the logical addresses used for swapping out theprocess
  3. The addressing mechanism must keep track of the physical addresses of the process, as well as thelogical addresses used for swapping out the process
  4. None of the others

# The purpose of a TLB is:

* 1. To cache page translation information
  2. To cache frequently used data
  3. To hold register values while a process is waiting to be run

To hold the start and length of the page table

1. What is Address Binding?
   1. going to an address in memory
   2. locating an address with the help of another address
   3. binding two addresses together to form a new address in a different memory space
   4. a mapping from one address space to another
2. Binding of instructions and data to memory addresses can be done at
   1. Compile time
   2. Load time
   3. Execution time
   4. All of the mentioned
3. If the process can be moved during its execution from one memory segment to another, then binding must be
   1. delayed until run time
   2. preponed to compile time
   3. preponed to load time
   4. none of the mentioned
4. What is Dynamic loading?
   1. loading multiple routines dynamically
   2. loading a routine only when it is called
   3. loading multiple routines randomly
   4. none of the mentioned
5. What is the advantage of dynamic loading?
   1. A used routine is used multiple times
   2. An unused routine is never loaded
   3. CPU utilization increases
   4. All of the mentioned
6. The idea of overlays is to
   1. data that are needed at any given time
   2. enable a process to be larger than the amount of memory allocated to it
   3. keep in memory only those instructions
   4. all of the mentioned
7. The must design and program the overlay structure.
   1. programmer
   2. system architect
   3. system designer
   4. none of the mentioned
8. The swaps processes in and out of the memory.
   1. Memory manager
   2. CPU
   3. CPU manager
   4. User
9. If binding is done at assembly or load time, then the process be moved to different locations after being swapped out and in again.
   1. Can
   2. Must
   3. can never
   4. may
10. In a system that does not support swapping
    1. the compiler normally binds symbolic addresses (variables) to relocatable addresses
    2. the compiler normally binds symbolic addresses to physical addresses
    3. the loader binds relocatable addresses to physical addresses
    4. binding of symbolic addresses to physical addresses normally takes place during execution
11. Which of the following is TRUE?
    1. Overlays are used to increase the size of physical memory
    2. Overlays are used to increase the logical address space
    3. When overlays are used, the size of a process is not limited to the size of the physical memory
    4. Overlays are used whenever the physical address space is smaller than the logical address space
12. The address generated by the CPU is referred to as
    1. Physical address
    2. Logical address
    3. Neither physical nor logical
    4. None of the mentioned
13. The address loaded into the memory address register of the memory is referred to as
    1. Physical address
    2. Logical address
    3. Neither physical nor logical
    4. None of the mentioned
14. The run time mapping from virtual to physical addresses is done by a hardware device called the
    1. Virtual to physical mapper
    2. Memory management unit
    3. Memory mapping unit
    4. None of the mentioned
15. The base register is also known as the
    1. basic register
    2. regular register
    3. relocation register
    4. delocation register
16. The size of a process is limited to the size of
    1. physical memory
    2. external storage
    3. secondary storage
    4. none of the mentioned
17. If execution time binding is being used, then a process be swapped to a different memory space.
    1. has to be
    2. can never
    3. must
    4. may
18. Swapping requires a
    1. Motherboard
    2. Keyboard
    3. Monitor
    4. backing store
19. The backing store is generally a
    1. fast disk
    2. disk large enough to accommodate copies of all memory images for all users
    3. disk to provide direct access to the memory images
    4. all of the mentioned
20. The consists of all processes whose memory images are in the backing store or in memory and are ready to run.
    1. wait queue
    2. ready queue
    3. cpu
    4. secondary storage
21. The time in a swap out of a running process and swap in of a new process into the memory is very high.
    1. context – switch
    2. waiting
    3. execution
    4. all of the mentioned
22. The major part of swap time is time.
    1. Waiting
    2. Transfer
    3. Execution
    4. none of the mentioned
23. Swapping be done when a process has pending I/O, or has to execute I/O operations only into operating system buffers.
    1. must
    2. can
    3. must never
    4. maybe
24. Swap space is allocated
    1. as a chunk of disk
    2. separate from a file system
    3. into a file system
    4. all of the mentioned
25. CPU fetches the instruction from memory according to the value of
    1. program counter
    2. status register
    3. instruction register
    4. program status word
26. A memory buffer used to accommodate a speed differential is called
    1. stack pointer
    2. cache
    3. accumulator
    4. disk buffer
27. Which one of the following is the address generated by CPU?
    1. physical address
    2. absolute address
    3. logical address
    4. none of the mentioned
28. Run time mapping from virtual to physical address is done by
    1. Memory management unit
    2. CPU
    3. PCI
    4. None of the mentioned
29. Memory management technique in which system stores and retrieves data from secondary storage for use in main memory is called?
    1. fragmentation
    2. paging
    3. mapping
    4. none of the mentioned
30. The address of a page table in memory is pointed by
    1. stack pointer
    2. page table base register
    3. page register
    4. program counter
31. Program always deals with
    1. logical address
    2. absolute address
    3. physical address
    4. relative address

.

1. The page table contains
   1. base address of each page in physical memory
   2. page offset
   3. page size
   4. none of the mentioned
2. What is compaction?
   1. a technique for overcoming internal fragmentation
   2. a paging technique
   3. a technique for overcoming external fragmentation
   4. a technique for overcoming fatal error
3. Operating System maintains the page table for
   1. each process
   2. each thread
   3. each instruction
   4. each address
4. The main memory accommodates
   1. operating system
   2. cpu
   3. user processes
   4. all of the mentioned

.

1. What is the operating system?
   1. in the low memory
   2. in the high memory
   3. either low or high memory (depending on the location of interrupt vector)
   4. none of the mentioned
2. In contiguous memory allocation
   1. each process is contained in a single contiguous section of memory
   2. all processes are contained in a single contiguous section of memory
   3. the memory space is contiguous
   4. none of the mentioned
3. The relocation register helps in
   1. providing more address space to processes
   2. a different address space to processes
   3. to protect the address spaces of processes
   4. none of the mentioned
4. With relocation and limit registers, each logical address must be the limit register
   1. less than
   2. equal to
   3. greater than
   4. none of the mentioned
5. The operating system and the other processes are protected from being modified by an already running process because
   1. they are in different memory spaces
   2. they are in different logical addresses
   3. they have a protection algorithm
   4. every address generated by the CPU is being checked against the relocation and limit registers
6. Transient operating system code is code that
   1. is not easily accessible
   2. comes and goes as needed
   3. stays in the memory always
   4. never enters the memory space
7. Using transient code, the size of the operating system during program execution.
   1. Increases
   2. Decreases
   3. Changes
   4. maintains
8. When memory is divided into several fixed sized partitions, each partition may contain
   1. exactly one process
   2. at least one process
   3. multiple processes at once
   4. none of the mentioned
9. In fixed size partition, the degree of multiprogramming is bounded by
   1. the number of partitions
   2. the CPU utilization
   3. the memory size
   4. all of the mentioned
10. The first fit, best fit and worst fit are strategies to select a
    1. process from a queue to put in memory
    2. processor to run the next process
    3. free hole from a set of available holes
    4. all of the mentioned
11. In internal fragmentation, memory is internal to a partition and
    1. is being used
    2. is not being used
    3. is always used
    4. none of the mentioned
12. A solution to the problem of external fragmentation is
    1. compaction
    2. larger memory space
    3. smaller memory space
    4. none of the mentioned
13. Another solution to the problem of external fragmentation problem is to
    1. permit the logical address space of a process to be noncontiguous
    2. permit smaller processes to be allocated memory at last
    3. permit larger processes to be allocated memory at last
    4. all of the mentioned

t

1. If relocation is static and is done at assembly or load time, compaction
   1. cannot be done
   2. must be done
   3. must not be done
   4. can be done
2. The disadvantage of moving all process to one end of memory and all holes to the other direction, producing one large hole of available memory is
   1. the cost incurred
   2. the memory used
   3. the CPU used
   4. all of the mentioned
3. is generally faster than and
   1. first fit, best fit, worst fit
   2. best fit, first fit, worst fit
   3. worst fit, best fit, first fit
   4. none of the mentioned
4. External fragmentation exists when?
   1. enough total memory exists to satisfy a request but it is not contiguous
   2. the total memory is insufficient to satisfy a request
   3. a request cannot be satisfied even when the total memory is free
   4. none of the mentioned
5. External fragmentation will not occur when?
   1. first fit is used
   2. best fit is used
   3. worst fit is used
   4. no matter which algorithm is used, it will always occur
6. Sometimes the overhead of keeping track of a hole might be
   1. larger than the memory
   2. larger than the hole itself
   3. very small
   4. all of the mentioned
7. When the memory allocated to a process is slightly larger than the process, then
   1. internal fragmentation occurs
   2. external fragmentation occurs
   3. both internal and external fragmentation occurs
   4. neither internal nor external fragmentation occurs
8. Physical memory is broken into fixed-sized blocks called
   1. Frames
   2. Pages
   3. backing store
   4. none of the mentioned
9. Logical memory is broken into blocks of the same size called
   1. frames
   2. pages
   3. backing store
   4. none of the mentioned
10. Every address generated by the CPU is divided into two parts. They are
    1. frame bit & page number
    2. page number & page offset
    3. page offset & frame bit
    4. frame offset & page offset
11. The is used as an index into the page table.
    1. frame bit
    2. page number
    3. page offset
    4. frame offset
12. The table contains the base address of each page in physical memory.
    1. process
    2. memory
    3. page
    4. frame
13. The size of a page is typically
    1. varied
    2. power of 2
    3. power of 4
    4. none of the mentioned
14. If the size of logical address space is 2 to the power of m, and a page size is 2 to the power of n addressing units, then the high order bits of a logical address designate the page number, and the low order bits designate the page offset.
    1. m, n
    2. n, m
    3. m – n, m
    4. m – n, n
15. With paging there is no
    1. Internal
    2. External
    3. either type of
    4. none of the mentioned
16. The operating system maintains a

fragmentation.

table that keeps track of how many frames have been

allocated, how many are there, and how many are available.

* 1. page
  2. mapping
  3. frame
  4. memory

1. Paging increases the time.
   1. waiting
   2. execution
   3. context – switch
   4. all of the mentioned
2. Smaller page tables are implemented as a set of
   1. Queues
   2. Stacks
   3. Counters
   4. registers
3. The page table registers should be built with
   1. very low speed logic
   2. very high speed logic
   3. a large memory space
   4. none of the mentioned
4. For larger page tables, they are kept in main memory and a points to the page table
   1. page table base register
   2. page table base pointer
   3. page table register pointer
   4. page table base
5. For every process there is a
   1. page table
   2. copy of page table
   3. pointer to page table
   4. all of the mentioned
6. Time taken in memory access through PTBR is
   1. extended by a factor of 3
   2. extended by a factor of 2
   3. slowed by a factor of 3
   4. slowed by a factor of 2
7. Each entry in a translation lookaside buffer (TLB) consists of
   1. key
   2. value
   3. bit value
   4. constant
8. An uniquely identifies processes and is used to provide address space protection for that process.
   1. address space locator
   2. address space identifier
   3. address process identifier
   4. none of the mentioned
9. Memory protection in a paged environment is accomplished by
   1. protection algorithm with each page
   2. restricted access rights to users
   3. restriction on page visibility
   4. protection bit with each page
10. When the valid – invalid bit is set to valid, it means that the associated page
    1. is in the TLB
    2. has data in it
    3. is in the process’s logical address space
    4. is the system’s physical address space
11. Illegal addresses are trapped using the bit.
    1. error
    2. protection
    3. valid – invalid
    4. access
12. When there is a large logical address space, the best way of paging would be
    1. not to page
    2. a two level paging algorithm
    3. the page table itself
    4. all of the mentioned
13. In paged memory systems, if the page size is increased, then the internal fragmentation generally
    1. becomes less
    2. becomes more
    3. remains constant
    4. none of the mentioned
14. How are jobs processed in a single-user system?
    1. sequentially
    2. randomly
    3. intermittently
    4. the longest job is processed first
15. What is another name for a fixed partition?
    1. complete partition
    2. direct partition
    3. static partition
    4. sized partition
16. What is the first step in the algorithm to load a job in a fixed partition?
    1. Compare job size to size of largest partition
    2. Determine job’s requested memory size
    3. Set counter to 1
    4. No partition available at this time, put job in waiting queue
17. In the partition scheme, the Memory Manager uses a table to keep track of jobs. What are the components of the table?
    1. partition size, memory address, status
    2. status, access, memory address
    3. partition size, status, access
    4. partition size, memory address, access, status
18. When does a fixed partition scheme work well?
    1. when jobs have the same size
    2. when job sizes are not known in advance
    3. when jobs have different sizes
    4. when all jobs are under 100K
19. What is the name for fragments of free memory between blocks of allocated memory?
    1. inefficient fit
    2. external fragmentation
    3. indirect partitioning
    4. internal fragmentation
20. Which memory allocation scheme places jobs in the first partition fitting the requirements?
    1. fixed partitioning
    2. dynamic fit memory allocation
    3. first-fit memory allocation
    4. best-fit memory allocation
21. Which memory allocation scheme results in the smallest amount of wasted space?
    1. fixed partitioning
    2. dynamic fit memory allocation
    3. first-fit memory allocation
    4. best-fit memory allocation
22. Consider the following space requirements for jobs 1-4 and memory blocks. Assuming a first-fit scheme is used, which job is not able to run?

Jobs:

J1 10K

J2 20K

J3 30K

J4 10K

Blocks:

B1 30K B2 15K B3 50K B4 20K

* 1. J1
  2. J3
  3. J2
  4. J4

1. Consider the following space requirements for jobs 1-4 and memory blocks. Assuming a best-fit scheme is used, which job is placed in the last block?

Jobs:

J1 10K

J2 20K

J3 30K

J4 10K 99999

Blocks:

B1 30K B2 15K B3 50K B4 20K

* 1. J1
  2. J3
  3. J2
  4. J4

1. The following algorithm can be described as --.
2. Set counter to 1
3. Do while counter <= number of blocks in memory If job\_size > memory\_size(counter)

Then counter = counter + 1 Else

load job into memory\_size(counter) adjust free/busy memory lists

go to step 4 End do

1. Put job in waiting queue
2. Go fetch next job
3. first-fit memory allocation
4. least-fit memory allocation
5. best-fit memory allocation
6. fixed partition memory allocation
7. The following algorithm can be described as --.
8. Initialize memory\_block(0) = 99999
9. Compute initial\_memory\_waste = memory\_block(0) – job\_size
10. Initialize subscript = 0
11. Set counter to 1
12. Do while counter <= number of blocks in memory If job\_size > memory\_size(counter)

Then counter = counter + 1 Else

memory\_waste = memory\_size(counter) – job\_size If initial\_memory\_waste > memory\_waste

Then subscript = counter initial\_memory\_waste = memory\_waste counter = counter + 1

End do

1. If subscript = 0

Then put job in waiting queue Else

Load job into memory\_size(subscript) adjust free/busy memory lists

1. Go fetch next job
2. first-fit memory allocation
3. least-fit memory allocation
4. best-fit memory allocation
5. fixed partition memory allocation
6. Assume the Memory Manager receives a request for a block of 200. When the best-fit algorithm is used, what is the beginning address of the block granted by the Memory Manager?

|  |  |
| --- | --- |
| Beginning Address | Memory Block Size |
| 4075 | 105 |
| 5225 | 5 |
| 6785 | 600 |
| 7560 | 20 |
| 7600 | 205 |
| 10250 | 4050 |
| 15125 | 230 |
| 24500 | 1000 |
| A. 6785 |  |
| B. 10250 |  |
| C. 7600 |  |
| D. 15125 |  |

1. How does an operating system reclaim fragmented memory space?
   1. deallocation
   2. compaction
   3. redirection
   4. reallocation
2. The contains the value that must be added to each address referenced in the program

so it will be able to access the correct memory addresses after relocation.

* 1. busy list
  2. relocation register
  3. compaction monitor
  4. bounds register

1. When should compaction be performed?
   1. when a certain percentage of memory becomes busy
   2. only when there are jobs waiting to get in
   3. after a prescribed amount of time has elapsed
   4. Any of the above, depending on the system
2. What is the actual memory address for a job that starts at 18K? A. 1,800

B. 18,432

C. 18,000

D. 180,000

1. In a fixed partition scheme, what is the problem with partitions that are too large?
   1. small jobs will have to wait
   2. external fragmentation
   3. large jobs will have to wait
   4. wasted memory
2. A single user contiguous memory management system
   1. has the advantage of simplicity in writing operating systems
   2. has the disadvantage of wasting memory for most jobs
   3. both a and b
   4. neither a nor b
3. Static partitioning systems were instituted to allow
   1. non-contiguous loading of programs
   2. multiprogramming
   3. both a&b
   4. non of them
4. Given the following static allocation scheme:

|  |  |  |
| --- | --- | --- |
| Partition | Contents | Size |
| 1 | operating system | 32K |
| 2 | free | 32K |
| 3 | free | 64K |

Where will a job requiring 96K bytes of memory be stored?

* 1. partition 1+ partition 2 +partition 3
  2. partition 2+ partition 3
  3. partition 3
  4. none of the above

1. A static partitioned memory management system has a total of six partitions. If one is allocated to the operating system, this will allow a total of
   1. five user jobs
   2. six user jobs
   3. (25) user jobs
   4. (25) user jobs
2. Consider the case where the Free Memory Storage List is kept in ascending order by partition size. The attempt to place a job in the smallest free partition in which it will fit is called:
   1. First Fit
   2. Best Fit
   3. Worst Fit
   4. No Fit
3. Consider the case where the Free Memory Storage List is kept in order by memory address (regardless of the partition size). The attempt to place a job in the first free block in which it will fit is called:
   1. First Fit
   2. Best Fit
   3. Worst Fit
   4. No Fit
4. Given the following dynamic partition allocation tables, respond to the next 3 questions.

**Allocated Partition Table**

Size Address Status

|  |  |  |
| --- | --- | --- |
| 8K | 312K | in use |
| 32K | 320K | in use |
| -- | -- | no entry |
| 120K | 384K | in use |
| -- | -- | no entry |

## Free Area Table

Size Address Status

|  |  |  |
| --- | --- | --- |
| 32K | 352K | free |
| 520K | 504K | free |
| -- | -- | no entry |
| -- | -- | no entry |

1. If a job comes into the system requiring 8K of main memory, using a First Fit allocation scheme, it will be given space from the free area of size:
   1. 32K
   2. 520K
   3. 352K
   4. 504K
2. The resulting fragment will be of size:
   1. 8K
   2. 24K
   3. 32K
   4. 40K
3. If the jobs currently running in the first two partitions finish before a new job arrives, the free table will be augmented by a space that is of size (assume compaction)?
   1. 8K
   2. 24K
   3. 32K
   4. 40K
   5. 632K
4. Relocatable partitioned memory was designed to:
   1. utilize low core storage
   2. reduce external fragmentation of memory
   3. allow for flexible addressing mechanisms
   4. all of them
5. A difficulty of all relocatable partitioning schemes, regardless of addressing mechanism is:
   1. the difficulty of distinguishing number representing addresses from numbers representing arithmetic values
   2. the overhead involved in moving program instructions and data to new locations
   3. all the above
   4. none of the above
6. Use the following description of main memory to answer the next 3 questions.

|  |  |  |
| --- | --- | --- |
| **PARTITION**  **Number** | **Partition SIZE** | **JOB SIZE** |
| 1 | 8K | 1K |
| 2 | 32K | 9K |
| 3 | 64K | 19K |

1. The space wasted by the job in partition 2 is:
   1. 7K
   2. 9K
   3. 23K
   4. 45K
2. The job in partition 3 would best fit in:
   1. partition 1
   2. partition 2
   3. partition 3
   4. none of the above
3. The length of each partition is:
   1. fixed
   2. variable
4. Consider the following diagram to answer the next 2 questions. Program relocation would result in a contiguous free space of:

0

Operating System

|  |
| --- |
| Program A |
| Program B |
| Waste 10K |
| Program C |
| Waste 12K |
| Program D |

640K

* 1. 12K
  2. 22K
  3. 2K
  4. 120K



|  |  |
| --- | --- |
| 1. | D |
| 2. | D |
| 3. | A |
| 4. | B |
| 5. | B |
| 6. | B |
| 7. | D |
| 8. | A |
| 9. | C |
| 10. | A |
| 11. | C |
| 12. | B |
| 13. | A |
| 14. | B |
| 15. | C |
| 16. | A |
| 17. | D |
| 18. | D |
| 19. | D |
| 20. | A |
| 21. | A |
| 22. | B |
| 23. | C |
| 24. | A |
| 25. | A |
| 26. | B |
| 27. | C |

|  |  |
| --- | --- |
| 28. | A |
| 29. | B |
| 30. | B |
| 31. | A |
| 32. | A |
| 33. | B |
| 34. | A |
| 35. | A |
| 36. | C |
| 37. | A |
| 38. | C |
| 39. | A |
| 40. | D |
| 41. | B |
| 42. | C |
| 43. | A |
| 44. | A |
| 45. | C |
| 46. | B |
| 47. | A |
| 48. | A |
| 49. | A |
| 50. | A |
| 51. | A |
| 52. | A |
| 53. | D |
| 54. | B |

|  |  |
| --- | --- |
| 55. | A |
| 56. | A |
| 57. | B |
| 58. | B |
| 59. | B |
| 60. | C |
| 61. | B |
| 62. | D |
| 63. | B |
| 64. | C |
| 65. | C |
| 66. | D |
| 67. | B |
| 68. | A |
| 69. | A |
| 70. | D |
| 71. | A |
| 72. | B |
| 73. | D |
| 74. | C |
| 75. | C |
| 76. | B |
| 77. | B |
| 78. | A |
| 79. | C |
| 80. | B |
| 81. | D |

|  |  |
| --- | --- |
| 82. | A |
| 83. | A |
| 84. | C |
| 85. | D |
| 86. | B |
| 87. | C |
| 88. | A |
| 89. | C |
| 90. | C |
| 91. | B |
| 92. | B |
| 93. | D |
| 94. | B |
| 95. | B |
| 96. | C |
| 97. | B |
| 98. | D |
| 99. | A |
| 100. | B |
| 101. | A |
| 102. | 1 A, 2B,  3D |
| 103. | B |
| 104. | B |
| 105. | 1C, 2B,  3A |
| 106. | B |