**Chapter 1**

1. **What is not a main structural element of a computer system?**

● Processor

● Main Memory

● I/O Modules

● System Bus

● Operating system

1. **Which of the following registers are used by the processor to exchange data with memory?**

● MAR and MBR

● PC and IR

● Program status word

● I/OAR and I/OBR

1. **Which of the following registers are used by the processor to exchange data with input/output module?**

● MAR and MBR

● PC and IR

● Program status word

● I/OAR and I/OBR

1. **Which of the following element of a computer system controls the operation of the computer and performs its data processing functions?**

● System bus

● Main memory

● I/O modules

● Processor

1. **Which of the following element of a computer system stores data and programs?**

* Main memory
* System bus
* Processor
* I/O modules

1. **Which of the following provides for communication among elements of a computer system?**

* Main memory
* System bus
* Processor
* I/O modules

1. **Which of the following element of a computer system moves data between the computer and its external environment?**

* Main memory
* Processor
* System bus
* *I/O modules*

1. **The processor contains a single data register, called**

* PSW
* PC
* IR
* AC

1. **This register specifies a particular input/output device**

* Memory address register
* I/OAR
* Memory buffer register
* I/OBR

1. **This register is used for the exchange of data between an I/O module and the processor**

* Memory address register
* I/OAR
* Memory buffer register
* I/OBR

1. **This register contains the data to be written into memory or which receives the data read from memory**

* I/OAR
* *memory buffer register*
* memory address register
* I/OBR

1. **This register specifies the location in memory for the next read or write**

* *Memory address register*
* I/OAR
* Memory buffer register
* I/OBR

1. **Index register, segment register, and stack register are example of:**

* Address register

1. **Which register contains the address of the next instruction to be fetched?**

* instruction register
* execution register
* program counter
* process counter

1. **Which register contains the instruction most recently fetched?**

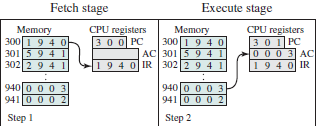
* instruction register
* execution register
* program counter
* process counter

1. **Which register contains condition codes set by the processor hardware as the result of operations?**

* Program counter
* Accumulator
* Program status word
* Stack pointer

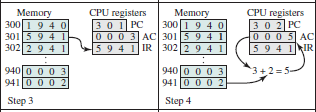
1. **The figure 1.4.**

**1. What is the address of the instruction that is being executed?**



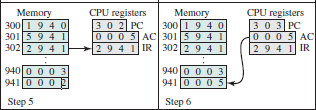
* 301
* 940
* 302
* 941

**2. What is the address of the instruction that is being executed?**



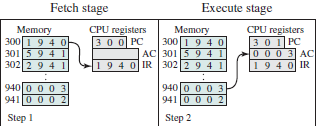
* 301
* 940
* 302
* 941

**3. What is the address of the instruction that is being executed?**



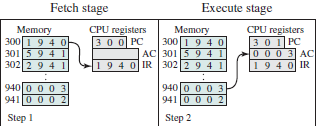
* 301
* 940
* 302
* 941

**4. What is the address of the instruction that will be executed next?**



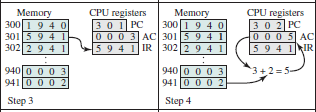
* 301
* 940
* 302
* 941

**5. Instruction in the IR will:**



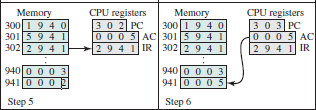
* Load 3 to AC from the memory
* Add 3 to AC from the memory
* Store 3 to AC from the memory
* Add 2 to AC from the memory

**6. Instruction in the IR will:**



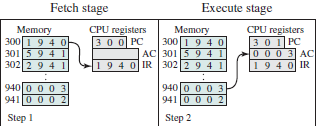
* Load 3 to AC from the memory
* Add 3 to AC from the memory
* Store 3 to AC from the memory
* Add 2 to AC from the memory

**7. Instruction in the IR will:**



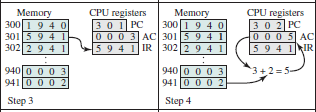
* Store content of AC to the memory
* Add 2 to the memory location from AC
* Store 5 to AC from the memory
* Add 3 to the memory location from AC

**8. What is the memory location of the data that is addressed in the instruction?**



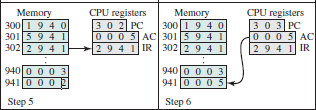
* 301
* 940
* 302
* 941

**9. What is the memory location of the data that is addressed in the instruction?**



* 301
* 940
* 302
* 941

**10. What is the memory location of the data that is addressed in the instruction?**



* 301
* 940
* 302
* 941

1. **The fetched instruction is loaded into the**

* IR
* Accumulator
* Memory
* PC

1. **At the beginning of each instruction cycle, the processor fetches an instruction from the memory. The address of the instruction is held in**

* IR
* MBR
* MAR
* PC

1. **The processor is executing ‘Load AC from memory’ instruction. Choose the correct micro-instructions:**

* PC -> MAR

M -> MBR

MBR -> AC

IR -> MAR

M -> MBR

MBR -> AC

* PC -> MBR

M -> MAR

MAR -> IR

IR -> MAR

M -> MBR

MBR -> AC

* PC -> MAR

M -> MBR

MBR -> IR

IR -> MAR

M -> MBR

MBR -> AC

* PC -> MAR

M -> MBR

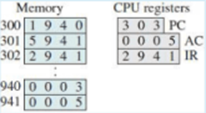
MBR -> IR

IR -> MBR

M -> MAR

MAR -> AC

1. **Choose the micro-instructions that reflect instruction execution stage shown in the figure below: (четыре вопроса в одном)**



* PC -> MAR 303

M -> MBR 2941

MBR -> IR 2941

* IR -> MAR 303

M -> MBR 0005

MBR -> M 0005

* IR -> MAR 941

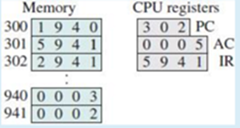
AC -> MBR 0005

MBR -> M 0005

* IR -> MAR 2941

AC -> MBR 0005

MBR -> M 0005



* IR -> MAR 941

M -> MBR 0002

MBR ->AC 0005

* PC -> MAR 301

M -> I/OBR 5941

I/OBR 5941

* IR -> MAR 941

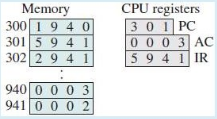
M -> MBR 0002

MBR -> AC 0002

* PC -> MAR 302

M -> MBR 2941

MBR -> IR 2941



* PC -> MAR 301

M -> MBR 6941

MBR -> IR 5941

* PC -> MAR 301

M -> MBR 5941

MBR -> IR 5941

* IR -> MAR 941

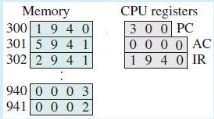
M -> MBR 0002

MBR -> AC 0002

* IR -> MAR 941

AC -> MBR 0003

MBR -> M 0003



* PC -> MAR 300

M -> MAR 1940

MBR -> IR 1940

* PC -> MAR 301

M -> MBR 1940

MBR -> IR 1940

* M -> MAR 300

M -> MBR 1940

MBR -> IR 1940

* PC -> MAR 300

M -> MBR 1940

MBR -> IR 1940

1. **When an I/O device completes an I/O operation, the device issues an interrupt signal to the processor and then:**

* The processor stops execution of the current instruction without finishing it and responds to the interrupt
* The processor loads the program counter with the entry location of the interrupt-handling routine
* The processor saves information needed to resume the current program at the point if interrupt
* The processor finishes execution of the current instruction before responding to the interrupt

1. **When the time required for the I/O operation is less that the time to complete the execution of instructions between write operations in the user program, it is:**

* Fast I/O wait
* Slow I/O wait
* Long I/O wait
* Short I/O wait

1. **When the time required for the I/O operation will take much more time than executing a sequence of user instructions, it is:**

* Slow I/O wait
* Long I/O wait
* Fast I/O wait
* Short I/O wait

1. **Most I/O devices are:**

* much slower than the processor

1. **If there no interrupts, after each write operation, the processor must:**

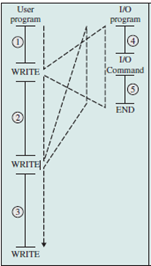
* Load new PC value
* Finish execution of current instruction
* Save the PSW and PC onto control stack
* Pause and remain idle until the I/O operation finishes

1. **The processor determined that there is a pending interrupt and sent an acknowledgement signal to the device that issued the interrupt. Then, the processor:**

The processor tests for a pending interrupt request, determines that there is one, and sends an acknowledgment signal to the device that issued the interrupt.The acknowledgment allows the device to remove its interrupt signal.

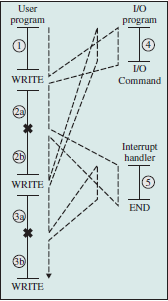
1. **The figure 1.5, 1.8, 1.9, 1.12, 1.13, 1.14, 1.19: (много вопросов в одном)**

**1. The following figure demonstrates:**



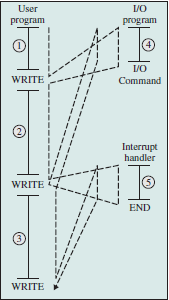
* Interrupts; fast I/O wait
* Interrupts; slow I/O wait
* *No interrupts*
* Interrupts; long I/O wait
* Interrupts; short I/O wait

**2. The following figure demonstrates:**



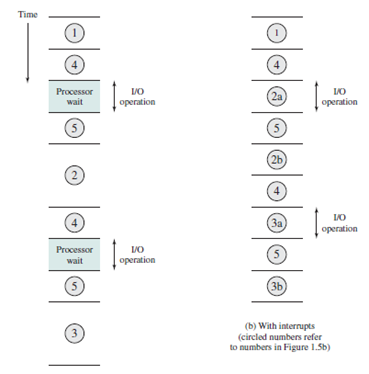
* Interrupts; fast I/O wait
* Interrupts; slow I/O wait
* No interrupts
* Interrupts; long I/O wait
* Interrupts; short I/O wait

**3. The following figure demonstrates:**



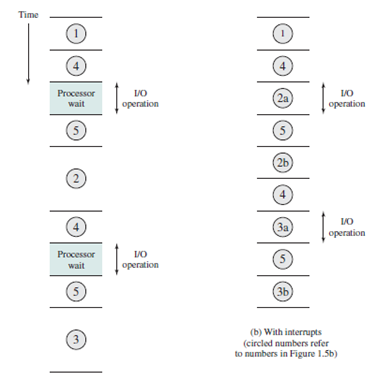
* Interrupts; fast I/O wait
* Interrupts; slow I/O wait
* No interrupts
* Interrupts; long I/O wait
* Interrupts; short I/O wait

**4. The figure a) demonstrates the program timing diagram for:**



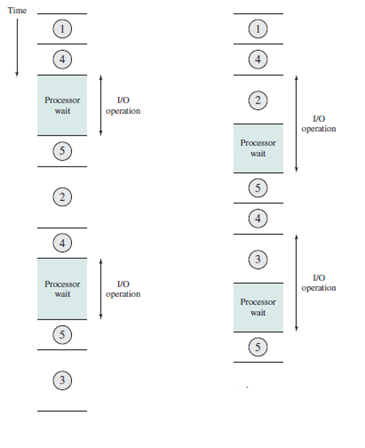
* Interrupts; fast I/O wait
* Interrupts; slow I/O wait
* *No interrupts*
* Interrupts; long I/O wait
* Interrupts; short I/O wait

**5. The figure b) demonstrates the program timing diagram for:**



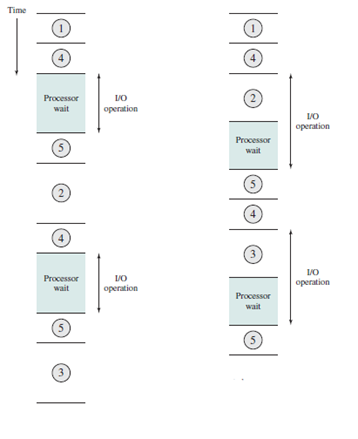
* Interrupts; fast I/O wait
* Interrupts; slow I/O wait
* No interrupts
* Interrupts; long I/O wait
* Interrupts; short I/O wait

**6. The figure a) demonstrates the program timing diagram for:**



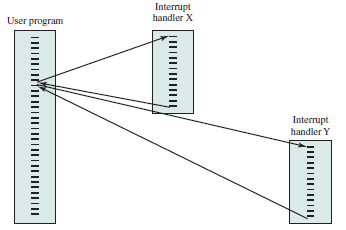
* Interrupts; fast I/O wait
* Interrupts; slow I/O wait
* *No interrupts*
* Interrupts; long I/O wait
* Interrupts; short I/O wait

**7. The figure b) demonstrates the program timing diagram for:**



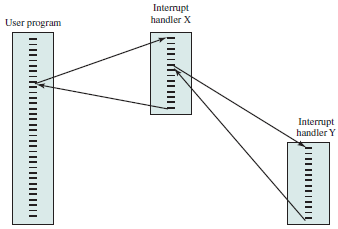
* Interrupts; fast I/O wait
* Interrupts; slow I/O wait
* No interrupts
* Interrupts; long I/O wait
* Interrupts; short I/O wait

**8. The figure below shows**



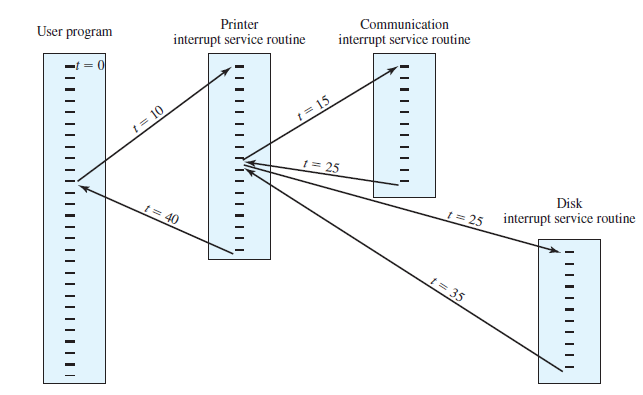
* Parallel interrupt processing
* Parent-child interrupt processing
* Sequential interrupt processing
* Nested interrupt processing

**9. The figure below shows**



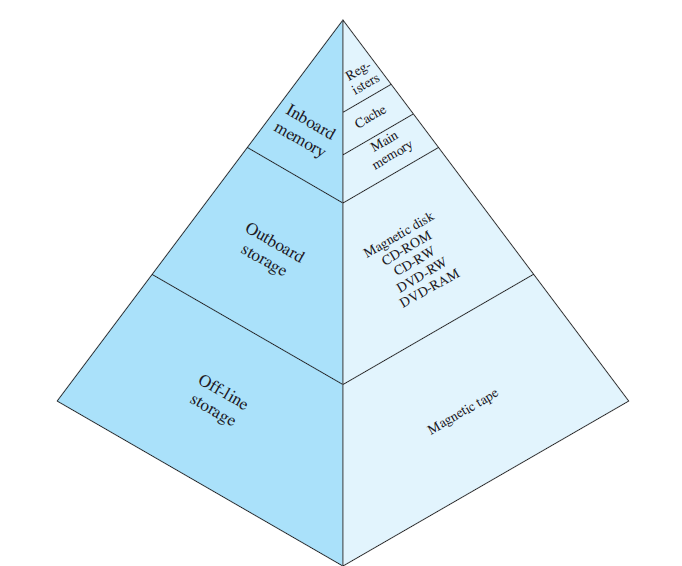
* Parallel interrupt processing
* Parent-child interrupt processing
* Sequential interrupt processing
* Nested interrupt processing

**10. The figure below shows**



* Parallel interrupt processing
* Time Sequence of Multiple Interrupts
* Sequential interrupt processing
* Nested interrupt processing

**11. As one goes down the memory hierarchy**

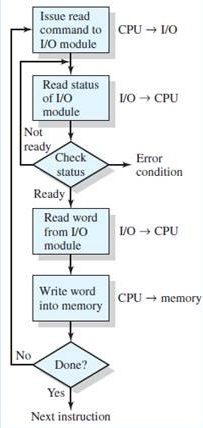
**the following occurs:**

* increasing frequency of access to the memory by the processor
* increasing cost per bit
* decreasing access time
* *increasing capacity*

**Memory hierarchy(пример вопроса как выше, ответы разные)**

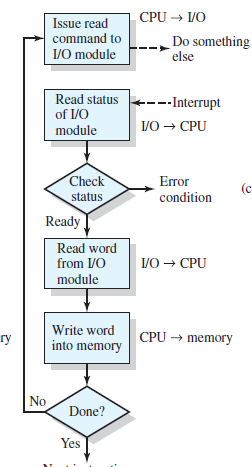
* *Decreasing cost/bit*
* *Increasing Capacity*
* *Increasing access time*
* *Decreasing frequency of the memory by the C.P.U.*

**12. The given flowchart**

 **is an example of:**

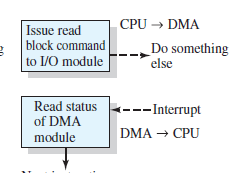
* Manual memory access
* Direct memory access (DMA)
* Interrupt-driven I/O
* Programmed I/O

**13. The given flowchart**

**is an example of:**

* Manual memory access
* Direct memory access (DMA)
* Interrupt-driven I/O
* Programmed I/O

**14. The given flowchart**

**is an example of:**

* Manual memory access
* Direct memory access (DMA)
* Interrupt-driven I/O
* Programmed I/O

1. **Complete the relationship concerning the memory systems: faster access time –**

* lower capacity
* *greater cost per bit*
* faster access speed
* smaller cost per bit

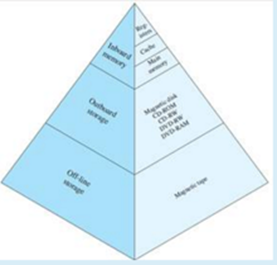
1. **Complete the relationship concerning the memory systems: greater capacity –**

* lower capacity
* greater cost per bit
* faster access speed
* *smaller cost per bit*

1. **Complete the relationship concerning the memory systems: greater capacity –**

* lower capacity
* slower access time
* greater cost per bit
* faster access speed

1. **As one goes down the memory hierarchy the following occurs:**



* increasing frequency of access to the memory by the processor
* increasing cost per bit
* decreasing access time
* increasing capacity

1. **The smaller, more expensive, faster memory is:**

* Disk drive
* Cache
* Main memory
* Register

1. **If the accessed word is found in the faster memory, that is defined as a:**

* bingo
* win
* *hit*
* evrika

1. **If the accessed word is not found in the faster memory, that is defined as a:**

* ricochet
* *miss*
* zero
* Loss

1. **This type of memory is nonvolatile:**

* Register
* Secondary memory
* Cache
* Main memory

1. **On all instruction cycles, the processor accesses memory at least once, to:**

to fetch the instruction, and often one or more additional times, to fetch operands and/or store results.

1. **When a processor attempts to read a byte or word of memory, it firstly checks the:**

* Register
* Secondary memory
* *Cache*
* Main memory

1. **Block size is:**

* The unit of data exchanged between cache and main memory
* Larger block size yields more hits until probability of using newly fetched data becomes less than the probability of reusing data that have to be moved out of cache
* (How much data should be transferred from main memory to cache)

1. **This determines which cache location the block will occupy:**

* Mapping function
* locality of reference
* replacement algorithm

1. **This chooses which block to replace when a new block is to be loaded into the cache and the cache already had all slots filled with other blocks:**

* *replacement algorithm*
* write policy
* cache size
* locality of reference

1. **If the contents of a block in the cache are altered, then it is needed to save it back to main memory before replacing it. This is called:**

* write policy
* mapping function
* locality of reference
* replacement algorithm

1. **Which of the following is not a technique possible for I/O operations?**

* Direct memory access (DMA)
* *Manual memory access*
* Interrupt-driven I/O
* Programmed I/O

1. **The processor issues an I/O command to a module and waits, periodically checking the status of the I/O module until it finds that the operation is complete.**

* Direct memory access (DMA)
* Manual memory access
* Interrupt-driven I/O
* *Programmed I/O*

1. **The I/O module performs the requested action and then sets the appropriate bits in the I/O status register but takes no further action to alert the processor.**

* Direct memory access (DMA)
* Manual memory access
* Interrupt-driven I/O
* *Programmed I/O*

1. **The described technique is:**

* Direct memory access (DMA)
* Manual memory access
* Interrupt-driven I/O
* *Programmed I/O*

1. **The main disadvantage of this technique: it is a time-consuming process that keeps the processor busy needlessly.**

* Direct memory access (DMA)
* Manual memory access
* Interrupt-driven I/O
* Programmed I/O

1. **The processor issues an I/O command to a module and then go on to some other useful work.**

* Direct memory access (DMA)
* Manual memory access
* *Interrupt-driven I/O*
* Programmed I/O

1. **The I/O module performs the requested action and alerts the processor to request service when it is ready to exchange data with the processor.**

* Direct memory access (DMA)
* Manual memory access
* *Interrupt-driven I/O*
* Programmed I/O

1. **The described technique is:**

* Direct memory access (DMA)
* Manual memory access
* *Interrupt-driven I/O*
* Programmed I/O

1. **The main disadvantage of this technique: consumes a lot of processor time, because every word of data that goes from memory to I/O module or from I/O module to memory must pass through the processor.**

* Direct memory access (DMA)
* Manual memory access
* *Interrupt-driven I/O*
* Programmed I/O

1. **The processor issues an I/O command to a separate module, by sending the type of operation (read/write), the address of the I/O device, the starting location in memory for data, number of words. Then the processor continues with other work.**

* Interrupt-driven I/O
* Programmed I/O
* Manual memory access
* Direct memory access (DMA)

1. **The module transfers the entire block of data to/from memory without going through the processor.**

* Interrupt-driven I/O
* Programmed I/O
* Manual memory access
* Direct memory access (DMA)

1. **The described technique is:**

* Interrupt-driven I/O
* Programmed I/O
* Manual memory access
* Direct memory access (DMA)

1. **The main disadvantage of this technique: there is a competition for bus usage.**

* Interrupt-driven I/O
* Programmed I/O
* Manual memory access
* Direct memory access (DMA)

**2 chapter**

1. **A hardcop y sign-up sheet was used to reserve computer time with**

* Time scheduling
* Time sliding
* Time slicing
* Time allocating

1. **A system clock generates interrupts at a rate of approximately one every 0.2 seconds. At each clock interrupt, the OS regained control and could assign the processor to another user. This technique is known as**

* Time scheduling
* Time sliding
* Time slicing
* Time allocating

1. **Batch operating systems were used (years)**

* From the late 1940s to the mid-1950s
* From the mid-1950 to early 1960s
* In early 1960s

1. **Job control language was used**

* In early 1960s used to provide instructions to the monitor,
* From late 1940s to the mid-1950s
* From the mid-1950s to early 1960s

1. **Serial processing was used**

* In early 1960s
* From late 1940s to the mid-1950s
* From the mid-1950s to early 1960s

1. **The central idea in this system was the use of a piece of software known as the monitor**

* Time-sharing-system
* Simple batch system
* Multiprogrammed batch system
* Serial processing

1. **The concept of a batch operating system was developed to**

* Maximize processor utilization (To improve utilization)
* Minimize response time
* Support interactive mode
* Minimize processor utilization

1. **The concept of modes of operation (user mode, system mode) was introduced with**

* Time-sharing-system
* Simple batch system
* Multiprogrammed batch system
* Serial processing

1. **The principal objective of the time-sharing system is to**

* Minimize response time
* Minimize processor utilization
* Maximize response time
* Maximize processor utilization

1. **The processor was often idle waiting for the I/O instruction to conclude. To overcome the problem the following approach was introduced: when one job needs to wait for I/O, the processor can switch to the other job. This was a**

* Multiprogramming or multitasking batch system
* Simple batch system
* Serial processing
* Time-sharing-system

1. **The programmer interacted directly with the computer hardware with**

* Simple batch systems
* Serial processing
* Time-sharing-systems
* Multiprogrammed batch systems

1. **The serial processing presented two main problems:**

* setup time
* using a monitor
* time slicing
* multiprogramming
* scheduling

1. **There were no OS with**

* Simple batch systems
* Serial processing
* Time-sharing-systems
* Multiprogrammed batch systems

1. **This system handles multiple interactive jobs**

* Time-sharing-system
* Simple batch system
* Serial processing
* Multiprogrammed batch system

1. **This system was developed by General Motors for the use on an IBM 701**

* Simple Batch Systems or GM-NAA I/O
* Serial processing
* Multiprogrammed batch system
* Time-sharing system

1. **This type of system was developed at MIT by a group known as Project MAC for the IBM 709 in 1961**

* Serial processing
* Multiprogrammed batch system
* Simple batch system
* Time-sharing-system

1. **'Lost signals, duplicate signals received'. This is**

* Improper synchronization
* Nondeterminate program operation
* Deadlocks
* Failed mutual exclusion

1. **'Two or more programs hung up waiting for each other'. This is**

* Falled mutual exclusions
* Deadlocks
* Nondeterminate program operation
* Improper synchronization

1. **'When programs share memory and their execution is interleaved by the processor, they may interfere with each other by overwriting common memory areas in unpredictable ways. The results of a particular program depend on the activities of other programs in a shared system'. This is**

* Falled mutual exclusions
* Deadlocks
* Nondeterminate program operation
* Improper syncronization

1. **Choose the storage management responsibility for "Many application programs require means for storing information for extended periods of time, after the computer has been powered down"**

* Long-term storage
* Protection and access control
* Automatic allocation and management
* Process isolation
* Support of modular programming

1. **Choose the storage management responsibility for "Programmers should be able to define program modules, and to create, destroy, and alter the size of modules dynamically"**

* Protection and access control
* Long-term storage
* Automatic allocation and management
* Process isolation
* Support of modular programming

1. **Choose the storage management responsibility for "The OS can achieve efficiency by assigning memory to jobs only as needed"**

* Protection and access control
* Long-term storage
* Automatic allocation and management
* Process isolation
* Support of modular programming

1. **Choose the storage management responsibility for "The OS must allow portions of memory to be accessible in various ways by various users"**

* Protection and access control
* Long-term storage
* Automatic allocation and management
* Process isolation
* Support of modular programming

1. **Choose the storage management responsibility for "The OS must prevent independent processes from interfering with each other's memory, both data and instructions"**

* Protection and access control
* Long-term storage
* Automatic allocation and management
* Process isolation
* Support of modular programming

1. **In operating system security and protection, authenticity:**

* Concerned with the proper verification of the identity of users and the validity of messages or data
* Assuring that users cannot read data for which access is unauthorized
* Protection of data from unauthorized modification
* Concerned with protecting the system against interruption

1. **In operating system security and protection, availability is:**

* Concerned with the proper verification of the identity of users and the validity of messages or data
* Assuring that users cannot read data for which access is unauthorized
* Protection of data from unauthorized modification
* Concerned with protecting the system against interruption

1. **In operating system security and protection, confidentiality is:**

* Concerned with the proper verification of the identity of users and the validity of messages or data
* Assuring that users cannot read data for which access is unauthorized
* Protection of data from unauthorized modification
* Concerned with protecting the system against interruption

1. **In operating system security and protection, data integrity is:**

* Concerned with the proper verification of the identity of users and the validity of messages or data
* Assuring that users cannot read data for which access is unauthorized
* Protection of data from unauthorized modification
* Concerned with protecting the system against interruption

1. **Short-term scheduler, or dispatcher, picks a process. Each process in the queue is given some time in turn. This strategy is called:**

* Round-robin technique
* Prioritizing
* Carousel technique
* Merry-go-round technique

1. **The base register**

* Defines the size of the region of (in bytes or words)
* Contains the index into the process list of the process currently controlling the processor
* Points to the next instruction in that process to be executed
* Contains the staring address of the region of memory occupied by the process

1. **The limit register**

* Defines the size of the region of (in bytes or words)
* Contains the index into the process list of the process currently controlling the processor
* Points to the next instruction in that process to be executed
* Contains the staring address of the region of memory occupied by the process

1. **The principal tool available to system programmers in developing the early multiprogramming and multiuser interactive systems**

* Interrupt

1. **The process index register**

* Contains the index into a process list of the process currently controlling the processor
* Define the region in memory occupied by the process
* Points to the next instruction in that process to be executed
* Contains the starting address of the region of memory occupied by the process

1. **This term was first used by the designers of Multics in the 1960s**

* Monitor
* Process
* Multiprogramming
* Time sharing

1. **Virtual address is a**

* Virtual address is a page number and an offset within the page

1. **'Ease of evolution of an OS' corresponds to the following objective:**

* Efficiency
* Convenience
* Ability to evolve

1. **'The OS as a resource manager' corresponds to the following objective:**

* Efficiency
* Convenience
* Ability to evolve

1. **'The OS as a User/Computer interface' corresponds to the following objective:**

* Efficiency
* Convenience
* Ability to evolve

1. **An application programs is developed by the**

* Application programmer
* End user
* Operating system
* Hardware

1. **The end user views a computer system in terms of**

* a set of machine instructions
* utilities
* a set of system programs
* a set of applications

1. **The hardware and software used in providing applications to a user can be viewed in a:**

* classified fashion
* layered fashion
* sandwich fashion
* leveled fashion

1. **This contains the most frequently used functions in the OS:**

* Centre
* Base
* Root
* Kernel (или nucleus)

1. **Utilities are**

* Application program
* System programs
* Opcodes
* Processor registers

**Chapter 3-4**

1. **Process is**

* a job in secondary memory
* a unit of activity characterized by execution of a sequence of instructions, a current state, and an associated set
* contents of main memory
* program in high level language kept on disk
* Program in High level language kept on disk
* Contents of main memory
* A job in secondary memory
* A program in execution

1. **A task in a blocked state**

* is running
* is executable
* must still be placed in the run queues
* is waiting for same temporarily unavailable resources

1. **The systems which allows only one process execution at a time, are called**

● unitasking systems

● uniprocessing systems

● uniprogramming systems

● unicasting systems

1. **The state of a process is defined by:**

* the activity is just executed by the process
* the final activity of the process
* the activity to next be executed by the process
* the current activity of the process

1. **Which of the following is not the state of a process?**

* new
* running
* terminated
* Old
* ready

1. **Suppose that a process is in “Blocked” state waiting for some I/O service. When the service is completed, it goes to the:**

* Ready state
* Running state
* Terminated state
* Suspended state

1. **Which of the following state transitions is not possible?**

* blocked to running
* running to blocked
* blocked to ready
* ready to running

1. **If a process is executing in its critical section, then no other processes can be executing in their critical section. This condition is called**

* critical exclusion
* asynchronous exclusion
* mutual exclusion
* synchronous exclusion

1. **Interprocess communication:**

* none of the above
* allows the processes to only synchronize their actions without communication
* allows processes to communicate and synchronize their actions when using the same address space
* allows processes to communicate and synchronize their actions without using the same address space

1. **A sequence of instructions, in a computer language, to get the desired result, is known as?**

* algorithm
* instruction
* process
* program

1. **In operating system, each process has its own:**

* all of the mentioned
* address space and global variables
* set of data
* program code

1. **For each process OS creates and manages:**

* Process control block
* Program code
* Program
* Thread control block

1. **Listing the sequence of instructions that are executed is called?**

* trace
* program counter
* instruction
* control block

1. **A process can be terminated due to:**

* fatal error
* normal exit
* all of the mentioned
* killed by another process

1. **What is the ready state of a process?**

* when process is scheduled to run after some execution
* none of the above
* when process is using CPU
* when process is unable to run until some task has been completed

1. **What is interprocess communication?**

* communication between two process
* communication between two threads of same process
* none of the above
* communication within the process

1. **A set of processes is deadlock if:**

* each process is blocked and will remain so forever
* none of the above
* all processes are trying to kill each other
* each process is terminated

1. **The address of the next instruction to be executed by the current process is provided by the:**

* pipe
* program counter
* CPU registers
* process stack

1. **A Process Control Block (PCB) doesn’t contain which of the following:**

* stack
* data
* process state
* bootstrap program
* code

1. **The Process Control Block is:**

* process type variable
* data structure
* block in memory
* secondary storage section

1. **The degree of multi-programming is:**

* the number of processes executed per unit time
* the number of processes in the I/O queue
* the number of processes in the ready queue
* the number of processes in memory

1. **A single thread of control allows the process to perform:**

* both
* only one task at a time
* multiple tasks at a time

1. **The objective of multi-programming is to:**

* have multiple programs waiting in a queue ready to run
* to maximize CPU utilization
* to minimize CPU utilization

1. **What is a long-term scheduler?**

* it selects which process has to be executed next and allocates CPU
* none of the above
* it selects which process has to be brought into the ready queue
* it selects which process has to remove from memory by swapping

1. **What is a medium-term scheduler?**

* it selects which process has to be executed next and allocates CPU
* none of the above
* it selects which process has to be brought into the ready queue
* it selects which process has to remove from memory by swapping

1. **What is a short-term scheduler?**

* it selects which process has to be executed next and allocates CPU
* none of the above
* it selects which process has to be brought into the ready queue
* it selects which process has to remove from memory by swapping

1. **The primary distinction between the short-term scheduler and the long-term scheduler is:**

* the length of their queues
* none of the above
* the type of processes the schedule
* the frequency of their execution

1. **In a multi-programming environment:**

* the processor executes more than one process at a time
* the programs are developed by more than one person
* more than one process resides in the memory
* a single user can execute many programs at the same time

1. **The context of a process in the PCB of a process DOESN’T contain:**

* context switch time
* the process state
* memory-management information
* the value of the CPU registers

1. **Which of the following state transitions is not possible?**

* blocked to running
* running to blocked
* blocked to ready
* ready to running

1. **Which process can be affected by other processes executing in the system?**

* init process
* cooperating process
* child process
* parent process

1. **Which one of the following is a synchronization tool?**

* socket
* thread
* pipe
* Semaphore

1. **Mutual exclusion can be provided by the:**

* none of the abovemaint
* mutex locks
* both
* binary semaphores

1. **When high priority task is indirectly preempted by medium priority task effectively inverting the relative priority of the two tasks, the scenario is called:**

* priority exchange
* priority inversion
* priority removal
* priority modification

1. **With ……………. only one process can execute at a time; meanwhile all other process are waiting for the processor. With ………….. more than one process can be running simultaneously each on a different processor.**

* Multiprocessing, multiprogramming
* Uniprogramming, Multiprocessing
* Multiprogramming, Uniprocessing
* Multiprogramming, Multiprocessing

1. **Interprocess communication allows:**

* allows process to synchronize activity
* is never necessary
* is usually done via disk drives
* is required for all processes

1. **Message passing system allows processes to:**

* share data
* communicate with one another without resorting to shared data
* name the recipient or sender of the message
* communicate with one another by resorting shared data

1. **The link between two processes P and Q to send and receive messages is called:**

* message-passing link
* all of the above
* communication link
* synchronization link

1. **In indirect communication between processes P and Q:**

● there is a mailbox to help communication between P and Q

● none of the above

● there is another machine between the two processes to help communication

● there is another process R to handle and pass on the messages between P and Q

1. **In the non-blocking send:**

* the sending process sends the message and resumes operation
* the sending process keeps sending until the message is received
* the sending process keeps sending until it receives a message
* none of the above

1. **In the blocking send, blocking receive:**

* both sender and receiver are blocked until message is delivered
* none of the above
* the sending process sends the message and resumes operation
* the sending process sends the message while receiver is blocked

1. **In the non-blocking send, blocking receive:**

* the sending process sends the message and resumes operation
* sender continues on, receiver is blocked until the requested message arrives
* the sending process keeps sending until it receives a message
* none of the above

1. **In the non-blocking send, non-blocking receive:**

* the sending process keeps sending until it receives a message
* neither of processes are required to wait
* the sending process keeps sending until the message is received
* the sending process sends the message and resumes operation

1. **Remote Procedure Calls (RPC) are used:**

* none of the above
* for communication between two processes on separate systems
* for communication between two processes on the same systems
* for communication between two processes remotely different from each other on the same system

1. **What is a trap/exception?**

* user generated interrupt caused by an error
* failure of the system
* hardware generated interrupt caused by an error
* software generated interrupt caused by an error

**Chapter 5**

1. **To avoid the race condition, the number of processes that may be simultaneously inside their critical section is:**

**●**  10

● 0

● 1

● 2

1. **The initial value of the semaphore that allows only one of the many processes to enter their critical sections, is?**

● 1

● 0

● 2

● 10

1. **Semaphores:**

● are used for memory management

● synchronize critical resources to prevent deadlock

● are used to do I/O

● synchronize critical resources to prevent contention

1. **Four necessary conditions for deadlock to exist are: mutual exclusion, no-preemption, circular wait and**

● race condition

● deadlock avoidance

● hold and wait

● starvation

1. **Part of a program where the shared memory is accessed and which should be executed invisibly, is called:**

● semaphores

● mutual exclusion

● critical section

● directory

1. **Banker's algorithm for resource allocation deals with:**

● deadlock prevention

● mutual exclusion

● deadlock recovery

● deadlock avoidance

1. **A situation where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which access takes place is called:**

● Entry section

● Race section

● Shared memory segments

● Process synchronization

1. **The segment of code in which the process may change common variables, update tables, write into files is known as?**

● critical section

● program

● non-critical section

● mutual exclusion

1. **Mutual exclusion means that?**

● if a process is executing in its critical section, then other processes must be executing in their critical section

● if a process is executing in its critical section, then all the resources of the system must be blocked until it finishes exclusion

● if a process is executing in its critical section, then no other process must be executing in their critical sections

● none of the above

1. **A minimum of \_\_\_\_\_ variable(s) is/are required to be shared between processes to solve the critical section problem?**

● two

● three

● one

● four

1. **An un-interruptible unit is known as:**

● single

● none of the above

● static

● atomic

1. **Semaphore is a/an \_\_\_\_\_\_\_ to solve the critical section problem?**

● special program for a system

● complex structure

● integer variable

● hardware for a system

1. **The two atomic operations permissible on semaphores are:**

● wait

● signal

● hold

● stop

1. **The code that changes the value of the semaphore is:**

● non-critical section code

● remainder section code

● none of the above

● critical section code

1. **The two kinds of semaphores are:**

● mutex

● counting

● binary

● decimal

1. **A binary semaphore is a semaphore with integer values :**

● 1

● -1

● 0

● 0.5

1. **A monitor is a type of:**

● none of the above

● high level synchronization construct

● low level synchronization construct

● semaphore

1. **What is the reusable resource:**

● none of the above

● that can be used by more than one process at a time

● that can be used by one process at a time and is not depleted by that use

● that can be shared between various threads

1. **Which of the following condition is required for deadlock to be possible?**

● mutual exclusion

● all of the above

● no resource can be forcibly removed from a process holding it

● a process may hold allocated resources while awaiting assignment of other resources

1. **A system is in the safe state if:**

● a) the system can allocate resources to each process in some order and still avoid a deadlock

● b) there exist a safe sequence

● both a and b

● none of the above

1. **Which one of the following is the deadlock avoidance algorithm?**

● dining philosophers problem

● elevator algorithm

● banker’s algorithm

● round-robin algorithm

1. **A problem encountered in multitasking when a process is perpetually denied necessary resources is called:**

● starvation

● deadlock

● aging

● inversion

1. **The number of resources requested by a process:**

● must exceed the total number of resources available in the system

● must always be equal to the total number of resources available in the system

● must not exceed the total number of resources available in the system

● must always be less than the total number of resources available in the system

1. **For non-sharable resources like a printer, mutual exclusion:**

● must exist

● none of the above

● must not exist

● may exist

1. **A deadlock avoidance algorithm dynamically examines the \_\_\_\_\_\_\_\_\_\_, to ensure that a circular wait condition can never exist.**

● resources

● operating system

● resource allocation state

● system storage state

1. **A state is safe, if:**

● the system can allocate resources to each process in some order and still avoid deadlock

● the system does not crash due to deadlock occurrence

● the state keeps the system protected and safe

● all of the above

**Chapter 6**

1. **The available vector defines**

* total amount of resources required for all processes
* total amount of each resource in the system
* total amount of each resource allocated to all processes
* total amount of each resource NOT allocated to any process

1. **The resource vector define**

* total amount of resources required for all processes
* total amount of each resource in the system
* total amount of each resource allocated to all processes
* total amount of each resource NOT allocated to any process

1. **This matrix defines the current allocation to process i of the resource j**

* Allocation matrix
* Resource matrix
* Request matrix
* Claim matrix

1. **This matrix defines the amount of resources of type j requested by process i**

* Allocation matrix
* Resource matrix
* Request matrix
* Claim matrix

1. **This matrix defines requirements of processes i for the resources j**

* Allocation matrix
* Resource matrix
* Request matrix
* Claim matrix

1. **The banker's algorithm is referred to as**

* denial of service
* resource allocation denial
* loan allocation denial
* resource allocation refusal

1. **'Hold and wait' condition for deadlock is defined as**

* Only one process may use a resource at a time
* No resource can be forcibly removed from process holding it
* Process holds a resource while awaiting for other resource
* Each process holds resource needed by next process in chain

1. **'Circular wait' condition for deadlock is defined as**

* Only one process may use a resource at a time
* No resource can be forcibly removed from process holding it
* Process holds a resource while awaiting for other resource
* Each process holds resource needed by next process in chain

1. **'No preemption' condition for deadlock is defined as**

* Only one process may use a resource at a time
* No resource can be forcibly removed from process holding it
* Process holds a resource while awaiting for other resource
* Each process holds resource needed by next process in chain

1. **'Mutual exclusion' condition for deadlock is defined as**

* Only one process may use a resource at a time
* No resource can be forcibly removed from process holding it
* Process holds a resource while awaiting for other resource
* Each process holds resource needed by next process in chain

1. **What is NOT an example of consumable resource?**

* Information in I\O buffers
* Signals
* Main memory
* Messages

1. **What is NOT an example of reusable resource?**

* semaphores
* processors
* interrupt
* i/o channels

1. **This resource can be created and destroyed**

* resumable resource
* restartable resource
* consumable resource
* finite resource

1. **This resource can be safely used by only one process at a time and is not depleted by that use**

* refurbished resource
* replenishing resource
* reusable resource
* refreshing resource

1. **Deadlock inevitable region can be referred to as a**

● fun region

● fatal region

● final region

● false region

1. **This illustrates the progress of two processes competing for two resources**

● joint process diagram

● joint ingres diagram

● joint regress diagram

● joint progress diagram

1. **When each process in the set is blocked awaiting for resource siezed by another blocked process**

● it is a deadend

● it is a deadlock

● it is a softlock

● it is a mortallock

**Chapter 7**

1. **Which of the following statements is false:**

* compaction does not involve relocation of programs
* the technique of storage compaction involves moving all occupied areas of storage to one end or other of main storage
* compaction is also known as garbage collection
* the system must stop everything while it performs the compaction

● I\O transfers are more efficient with large pages

● a large page size causes instructions and data that will not be referenced brought into primary storage

● a small page size causes large page tables

● internal fragmentation is increased with small pages

1. **Memory is:**

* is the device where information stored
* is a sequence of the instructions
* is a device that performs a sequence of operations specified by instructions in memory
* is typically characterized by interactive processing and time-slicing of the CPU’s time to allow quick response to each user

1. **Processor is:**

* is the device where information stored
* is a sequence of the instructions
* is a device that performs a sequence of operations specified by instructions in memory
* is typically characterized by interactive processing and time-slicing of the CPU’s time to allow quick response to each user

1. **Program is:**

* is the device where information stored
* is a sequence of the instructions
* is a device that performs a sequence of operations specified by instructions in memory
* is typically characterized by interactive processing and time-slicing of the CPU’s time to allow quick response to each user

1. **The memory allocation scheme subject to "external" fragmentation is?**

* segmentation
* swapping
* multiple contiguous fixed partitionings
* pure demand paging

1. **The memory allocation scheme subject to "internal" fragmentation is?**

* segmentation
* swapping
* multiple contiguous fixed partitionings
* pure demand paging

1. **Any program, no matter how small, will occupy an entire partition results in?**

* internal fragmentation
* segmentation
* paging
* external fragmentation

1. **How to solve problem of equal-size partitions?**

swapping (?) ----- page 318 book

* unequal-size partitions
* segmentation
* compaction
* virtual memory segmentation

1. **How to solve problem of "external" fragmentation?**

* compaction
* larger memory space
* smaller memory space
* none of these

1. **A page fault?**

* occurs when a program accesses a page of memory
* is a reference to a page belonging to another program
* is an access to a page not currently in memory
* is an error is a specific page

1. **Which of the following statements is false?**

* I\O transfers are more efficient with large pages
* a large page size causes instructions and data that will not be referenced brought into primary storage
* a small page size causes large page tables
* internal fragmentation is increased with small pages

1. **Swapping:**

* allows many programs to use memory simultaneously
* allows each program in turn to use the memory
* does not work with overlaying
* none of the above

1. **Which of the following is not true about the memory management?**

* virtual memory is used only in multi-user systems
* segmentation suffers from external fragmentation
* paging suffers from internal fragmentation
* segmented memory can be paged
* None of the above

1. **True or false: segmentation suffers from external fragmentation?**

**True**

1. **True or false: paging suffers from external fragmentation?**

**True**

1. **What is a method of memory allocation by which the program is subdivided into equal portions, or pages and core is subdivided into equal portions or blocks?**

* virtual memory
* segmentation
* paging
* partition

1. **In memory systems, boundary registers?**

* track the beginning and ending of programs
* track page boundaries
* are only necessary with fixed partitions
* are used for temporary program variable storage

1. **A relationship between processes such that each has some part (critical section) which must not be executed while the critical section of another is being executed, is known as?**

* Multitasking
* Semaphore
* Mutual exclusion
* multiprogramming

1. **CPU fetches the instruction from memory according to the value of?**

* status register
* program counter
* instruction register
* program status word

1. **A memory used to provide a high speed is called?**

* cache
* main memory
* disk buffer
* stack pointer

1. **Which one of the following is the address generated by CPU?**

* logical address
* physical address
* absolute address
* none of the above

1. **Runtime mapping from virtual to physical address is done by?**

* CPU
* none of the above
* memory management unit
* PCI

1. **The address of a page table in memory is pointed by:**

* page table base register
* stack pointer
* page register
* program counter

1. **Program always deals with:**

* logical address
* physical address
* relative address
* absolute address

1. **What is compaction?**

* a paging technique
* a technique for overcoming fatal error
* a technique for overcoming internal fragmentation
* a technique for overcoming external fragmentation

1. **Operating System maintains the page table for:**

* each process
* each thread
* each instruction
* each address

1. **In contiguous memory allocation:**

* each process is contained in a single contiguous section of memory
* none of the above
* the memory space is contiguous
* all processes is contained in a single contiguous section of memory

1. **When memory is divided into several fixed sized partitions, each partition may contain \_\_\_\_\_\_\_\_.**

* exactly one process
* none of the above
* at least one process
* multiple processes at once

1. **In fixed sized partition, the degree of multiprogramming is bounded by \_\_\_\_\_\_\_\_\_\_\_.**

* the memory size
* all of the above
* the number of partitions
* the CPU utilization

1. **The first fit, best fit and worst fit are strategies to select a \_\_\_\_\_\_.**

* process from a queue to put in memory
* all of the above
* free hole from a set of available holes
* processor to run the next process

1. **In internal fragmentation, memory is internal to a partition and:**

* is not being used
* none of the above
* is always used
* is being used

1. **A solution to the problem of external fragmentation is:**

* compaction
* larger memory size
* unequal size parts
* smaller memory space

1. **\_\_\_\_\_\_\_\_\_\_ is generally faster than \_\_\_\_\_\_\_\_\_ .**

* worst fit, best fit
* best fit, first fit
* first fit, best fit
* none of the above

1. **External fragmentation exists when:**

* none of the above
* the total memory is insufficient to satisfy a request
* enough total memory exists to satisfy a request but it is not contiguous
* a request cannot be satisfied even when the total memory is free

1. **External fragmentation will not occur when:**

* no matter which algorithm is used, it will always occur
* first fit is used
* best fit is used
* next fit is used

1. **When the memory allocated to a process is slightly larger than the process, then:**

* both will occur
* external fragmentation occurs
* internal fragmentation occurs
* none of the above

1. **Main memory is broken into fixed-sized blocks called \_\_\_\_\_\_\_\_.:**

* none of the above
* pages
* frames
* segments

1. **Fixed-length block of data in secondary memory is called \_\_\_\_\_\_\_\_.:**

* none of the above
* frames
* segments
* pages

1. **Variable-length block of data that resides in secondary memory is called \_\_\_\_\_\_\_\_.**

* none of the above
* frames
* segments
* pages

1. **Every address generated by the CPU is divided into two parts:**

* frame offset
* page number
* frame bit
* page offset

1. **The \_\_\_\_\_\_\_\_\_\_ is used as an index into the page table.**

* page number
* page offset
* frame bit
* frame offset

1. **The \_\_\_\_\_ table contains the base address of each page in physical memory.**

* process
* memory
* page
* frame

1. **With paging there is no \_\_\_\_\_\_\_\_ fragmentation.**

* either type of
* none of the above
* external
* internal

1. **The operating system maintains a \_\_\_\_\_\_ table that keeps track of how many frames have been allocated, how many are there, and how many are available.**

* memory
* page
* frame
* segment

1. **For every process there is a \_\_\_\_\_\_\_\_\_\_.**

* pointer to page table
* copy of page table
* frame table
* page table

1. **If a page number is not found in the TLB, then it is known as a:**

* TLB miss
* buffer miss
* TLB ht

1. **If a page table entry is present in the TLB, then it is known as a:**

* buffer miss
* TLB miss
* page fault
* TLB hit

1. **If a page table entry is not in main memory, then it is known as a:**

* page fault
* buffer miss
* TLB hit
* TLB miss

1. **In segmentation, each address is specified by (choose two):**

* a segment number
* an offset
* a value
* a key

**236. The set-associative map technique is a combination of the direct and associative technique**

1. Верно
2. Неверно

**237. The spatial aspect of the locally of reference means \_\_\_\_\_\_\_\_**

1. That the instruction in close proximity of the instruction executed will be executed in future
2. That the recently executed won’t be executed again
3. That the instruction executed will be executed at a later time
4. That the recently executed instruction is executed again next

**238. The technique of searching for a block by going through all the tags is \_\_\_\_\_\_**

1. Linear search
2. Associative search
3. indirect search
4. Binary search

**239. The memory blocks are mapped on to the cache with the help of \_\_\_\_\_\_\_\_\_**

1. Hash functions
2. Mapping functions
3. Replacement policy
4. Write policy

**240. For 4 page frames, the following is the reference string:**

**1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6**

**How many page faults does the LRU page replacement algorithm produce(including page faults when the frames were free)?**

1. 10
2. 14
3. 12
4. 8

**241. Consider a logical address space of 32 pages with 512 words per page, mapped onto a physical memory of 16 frames. How many bits are required in the physical address?**

1. 13
2. 15
3. 12
4. 14

**242. Assuming a 1-KB page size, for the following address reference(provided as decimal number) 256, what is the page number?**

**Answer: \_\_\_\_\_\_\_\_\_\_\_**

**243. At a particular time of computation the value of a counting semaphore is 7. Then 20P operations were completed on this semaphore. The resulting value of the semaphore is**

1. 7
2. 12
3. 42
4. 2

**244. If it is not possible to tell, by looking at the program, what will happen when it executes, then the program is**

1. non-divisible
2. non-deadlocked
3. non-deterministic
4. non-distributed

**245. The program follows to use a shared binary semaphore T:**

**Process A**

**int Y;**

**A1: Y=X\*2;**

**A2: X=Y;**

**signal(T);**

**Process B**

**int Z;**

**B1:wait(t);**

**B2: Z = X + 1;**

**X = Z;**

**T is set to 0 before either process begins execution and, as before, X is set to 5. Now, how many different values of X are possible after both processes finish executing?**

1. two
2. one
3. four
4. three

**246. Computer programmers are often concerned with synchronization constraints, which are requirements pertaining to the order of events. They are:**

1. Serialization and Mutual inclusion
2. Scheduling and dispatching
3. Queueing and dispatching
4. Serialization and Mutual exclusion

**247. The effectiveness of the cache memory is based on the property of \_\_\_\_\_\_**

1. Memory access time
2. Memory size
3. Memory localisation
4. Locality of reference

**248. In direct mapping the presence of the block in memory is checked with the help of \_\_\_\_\_\_\_ field.**

1. word
2. block
3. tag
4. ser

**249. This matrix gives the maximum requirement of each process for each resource**

1. Available matrix
2. Resources matrix
3. Claim matrix
4. Allocation matrix

**250. If the resource allocation graph contains a cycle**

1. then a deadlock exists
2. either deadlock exists or system is in a safe state
3. then the system is in a safe state
4. then a deadlock does not exist

**251. Segment replacement algorithms are more complex than page replacement algorithms because**

1. Segments have fixed size
2. Segments have variable sizes
3. Pages are better than segments
4. Segments are better than pages

**252. When does processor generate an Interrupt indicating a memory access fault?**

1. If it encounters a logical address that is not in secondary memory
2. If it encounters a logical address that is not in cache memory
3. If it encounters a logical address that is not in main memory
4. If there is no enough free memory for the running application

**253. Which of the replacement algorithms selects for replacement that page for which the time to the next reference is the longest?**

1. FIFO
2. optimal
3. Clock
4. LRU

**254. Given five memory partitions of 100KB, 500KB, 200KB, 300KB, and 600KB. The first-fit algorithm places processes of 212KB, 417KB, 112KB, and 426KB. In whch partition 426KB is placed?**

1. No sufficient amount
2. 600KB
3. 200KB
4. 500KB
5. 100KB
6. 300KB

**255. Given five memory partitions of 100KB, 500KB, 200KB, 300KB, and 600KB. The worst-fit algorithm places processes of 212KB, 417KB, 112KB, and 426KB. In whch partition 426KB is placed?**

1. 500KB
2. No sufficient amount
3. 600KB
4. 200KB
5. 100KB
6. 300KB

**256. For 4 page frames, the following is the reference string:**

**1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6**

**How many page faults does the Optimal page replacement algorithm produce (including page faults when the frames were tree)?**

1. 14
2. 8
3. 10
4. 12

**257. A computer system has 6 tape drives, with ‘n’ processes competing for them. Each process may need 3 tape drives. The maximum value of ‘n’ for which the system is guaranteed to be deadlock free is**

1. 4
2. 3
3. 1
4. 2

**258. A system has 3 processes sharing 4 resources. If each process needs a maximum of 2 units then**

1. deadlock may occur
2. deadlock must occur
3. they all will starve
4. deadlock can never occur

**259. The two kinds of semaphore are**

1. binary & counting
2. decimal & binary
3. mutex & counting
4. counting & decimal

260. The size of a process is limited to the size of

1. none of the mentioned
2. external storage
3. physical memory
4. secondary storage

261. In FIFO page replacement algorithm, when a page must be replaced

1. random page is chosen
2. oldest page is chosen
3. newest page is chosen
4. none of the mentioned

262. 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 base
4. page table register pointer

263. The essential content(s) in each entry of a page table is/are

1. Virtual page number
2. Access right information
3. Both virtual page number and page frame number
4. Page frame number

264. When a program tries to access a page that is mapped in address space but not loaded in physical memory, then

1. no error occurs
2. page fault occurs
3. segmentation fault occurs
4. fatal error occurs

265. A page fault occurs when

1. all of the mentioned
2. a page cannot be accessed due to its absence from memory
3. a page is invisible
4. a page gives inconsistent data

266. sem.wait()

1. increments the semaphore
2. initializes the semaphore
3. reads the semaphore
4. decrements the semaphore

267. Each entry in a segment table has a

1. none of the mentioned
2. segment value
3. segment peak
4. segment base

268. Virtual memory takes advantage of

1. Fast CPU
2. Large secondary memory
3. Modular programming
4. Scheduling

269. Because of virtual memory, the memory can be shared among

1. instructions
2. none of the mentioned
3. threads
4. processes

270. If no frames are free and modify bit is set for all pages in frames, \_\_\_\_\_ page transfer(s) is/are required

1. two
2. three
3. four
4. one

271. Which is the fastest cache mapping function?

1. Fully associative mapping
2. Direct mapping
3. Set associative mapping

272. This bit indicates if the slot holds a block belonging to the program being executed

1. Reference bit
2. Control bit
3. Valid bit
4. Dirty bit

273. The size of virtual storage is limited by….

1. the addressing scheme of the computer system and by the amount of secondary memory
2. The portion of a process that is actually in mean memory at any time
3. the number of processors in a system
4. The portion of a secondary memory that contains pages or segments of particular application
5. the actual number of main storage locations
6. The portion of a process that is in a secondary memory at any time
7. the amount of main memory and secondary memory
8. ss?
9. The set of physical addresses

274. Optimal page - replacement algorithm is difficult to implement, because

1. it requires a lot of information
2. it is extremely expensive
3. it requires future knowledge of the reference string
4. it is too complex

275. What does process page table entry contain?

1. the frame number of corresponding page in main memory
2. size of the process
3. physical address of the page in secondary memory
4. virtual address of the page that is used by program

276. When several processes access the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called:

1. race condition
2. essential condition
3. critical condition
4. dynamic condition

277.Given five memory partitions of 100KB, 500KB, 200KB, 300KB, and 600KB. The best-fit algorithm places processes of 212KB, 417KB, 112KB, and 426KB. In whch partition 112KB is placed?

1. 600KB
2. 200KB
3. 500KB
4. 100KB
5. 300KB
6. No sufficient amount

278. Address Binding is

1. going to an address in memory
2. a mapping from one address space to another
3. locating an address with the help of another address
4. binding two addresses together to form a new address in a different memory space

279. When free memory is split into many unconnected pieces we call it:

1. Memory distribution
2. External fragmentation
3. Internal fragmentation
4. Compaction

280. 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, m
2. n, m
3. m - n, n
4. m, n

281. **Assuming a 1-KB page size, for the following address reference(provided as decimal number) 2375, what is the page number?**

**Answer: \_\_\_\_\_\_\_\_**

282. A system has 12 magnetic tape drives and 3 processes: P0, P1, and P2. Process P0 requires 10 tape drives, P1 requires 4 and P2 requires 9 tape drives. Process Maximum needs Currently allocated

P0 10 5

P1 4 2

P2 9 2

Which of the following sequence is a safe sequence?

1. P1, P2, P0
2. P1, P0, P2
3. P2, P0, P1
4. P0, P1, P2

283. If there are 32 segments, each of size 1KB, then the logical address should have

1. 13 bits
2. 14 bits
3. 16 bits
4. 15 bits

284. A multilevel page table is preferred in comparison to a single level page table for translating virtual address to physical address because

1. It helps to reduce the number of page faults in page replacement algorithms
2. it is required by the translation lookaside buffer
3. it reduces the memory access time to read or write a memory location
4. it helps to reduce the size of page table needed to implement the virtual address space of a process

285. If the resources are always preempted from the same process, \_\_\_\_\_\_\_ can occur

1. deadlock
2. starvation Problems
3. system cash
4. aging

286. Which one of the following is a visual (mathematical) way to determine the deadlock occurrence?

1. starvation graph
2. deadlock occurrence graph
3. inversion graph
4. resource allocation graph

287. To ensure that the hold and wait condition never occurs in the system, it must be ensured that

1. whenever a resource is requested by a process, it is not holding any other resources
2. each process must request and be allocated all its resources before it begins its execution
3. a process can request resources only when it has none
4. all the mentioned

288. If no cycle exists in the resource allocation graph

1. all the mentioned
2. then the system will not be in a safe state
3. then the system will be in a safe state
4. none of the mentioned

289. The memory allocation scheme subject to “internal” fragmentation is?

1. segmentation
2. pure demand paging
3. virtual memory segmentation
4. fixed partitioning

290. Thrashing \_\_\_\_\_\_\_ the CPU utilization

1. keeps constant
2. decreases
3. none of the mentioned
4. increases

291. The data structures available in the Banker’s algorithm are (выберите один или несколько ответов)

1. allocation
2. need
3. all of the mentioned
4. available

292. Each request requires that the system consider the \_\_\_\_\_\_\_ to decide whether the current request can be satisfied or must wait to avoid a future possible deadlock

1. resources currently allocated to each process
2. processes that have previously been in the system
3. resources currently available
4. future requests and releases of each process

293. What is the drawback of banker’s algorithm

1. none of the mentioned
2. resources are always available
3. in advance processes rarely know that how much resource they will need
4. the number of processes does not change as time progresses

294. For a deadlock to arise, which of the following conditions must hold simultaneously?(выберите один или несколько ответов)

1. mutual exclusion
2. no preemption
3. all the mentioned
4. hold and wait

295. Memory partitioning algorithm that uses binary tree as a data structure is called

1. Buddy system
2. First-fit algorithm
3. Best-fit algorithm
4. Next-fit algorithm

296. A mutex

1. cannot be accessed by processes
2. can be accessed from multiple processes
3. must be accessed from only one process
4. is a hexademical value

297. \_\_\_\_\_\_\_\_ replacement allows each process to only select from its own set of allocated frames

1. local
2. public
3. global
4. universal

298. LRU page - replacement algorithm associates with each page the \_\_\_\_\_\_\_\_

1. page after and before it
2. all of the mentioned
3. the time of that page’s last use
4. time it was brought into memory

299. Which of the following is true?

1. Overlays are used whenever the physical address space is smaller than the logical address space
2. When overlays are used, the size of a process is not limited to the size of the physical memory
3. Overlays are used to increase the logical address space
4. Overlays are used to increase the size of physical memory

300. The bounded buffer problem is also known as

1. Producer - Consumer problem
2. Banker’s algorithm
3. Dining - Philosopher problem
4. Readers - Writers problem

301. Semaphore operation semV() is alternative to

1. sem.wait()
2. sem.signal()
3. semaphore read
4. semaphore initialization

302.