

SYSTEM MCQs COLLECTION



NSCT Prep

Free MCQ Practice for NSCT Test Preparation



Operating Systems

1080 Multiple Choice Questions

nsctprep.dev

This dataset is created and compiled by Muhammad Abdullah Awais

© 2026 NSCT Prep. All rights reserved.

Easy Questions

360 questions

Q1. What is an operating system?

- A. System software that manages hardware and resources
- B. A physical component installed inside the computer
- C. A programming language for writing source code
- D. An application designed for end-user productivity

Answer: A

Q2. Which of the following is NOT a function of an operating system?

- A. Process management
- B. File system management
- C. Compiling user programs
- D. Memory management

Answer: C

Q3. Which of the following is an example of a mobile operating system?

- A. Android mobile OS
- B. Ubuntu desktop OS
- C. MS-DOS legacy OS
- D. Windows Server OS

Answer: A

Q4. What does the term 'booting' mean in the context of operating systems?

- A. Shutting down the computer safely
- B. Loading the OS into main memory
- C. Formatting the hard disk drive
- D. Installing new software packages

Answer: B

Q5. In a batch operating system, jobs are processed:

- A. Sequentially in submitted order
- B. Based on assigned priority
- C. Randomly without any order
- D. Simultaneously on all cores

Answer: A

Q6. Which type of operating system allows multiple users to use the computer simultaneously?

- A. Single-user OS
- B. Embedded OS
- C. Batch-job OS
- D. Time-sharing OS

Answer: D

Q7. The interface between the user and the operating system is called:

- A. The BIOS
- B. The API
- C. Shell
- D. Kernel

Answer: C

Q8. Which of the following is a real-time operating system characteristic?

- A. It does not support multitasking at all
- B. It only supports one single user at once
- C. It processes all jobs in large batches
- D. It guarantees response within strict time

Answer: D

Q9. UNIX was originally developed at:

- A. IBM Corporation
- B. Apple Computer
- C. AT&T Bell Labs
- D. Microsoft Research

Answer: C

Q10. The core component of an operating system is called the:

- A. Terminal
- B. Kernel
- C. Desktop
- D. Shell

Answer: B

Q11. Which of the following is a type of system call?

- A. Compiler optimization
- B. Syntax validation
- C. Process control
- D. Code refactoring

Answer: C

Q12. In a layered OS structure, the innermost layer is:

- A. Hardware layer
- B. Applications
- C. File systems
- D. User interface

Answer: A

Q13. Which of the following is an example of a command-line interpreter?

- A. Microsoft Word
- B. Bash shell
- C. Google Chrome
- D. Windows Explorer

Answer: B

Q14. System programs provide:

- A. Direct access to hardware resources
- B. Environment for program development
- C. Only file management service calls
- D. Gaming and multimedia functionality

Answer: B

Q15. The OS component that provides a graphical way to interact with the computer is called:

- A. SDK
- B. CLI
- C. API
- D. GUI

Answer: D

Q16. Which system call is used to create a new process in UNIX?

- A. exec()
- B. open()
- C. fork()
- D. wait()

Answer: C

Q17. A monolithic kernel is one where:

- A. The entire OS runs as one kernel program
- B. The OS is divided into separate modules
- C. Only the scheduler runs in kernel mode
- D. Each function runs as a separate process

Answer: A

Q18. Which of the following manages how data is stored and retrieved on a disk?

- A. Network manager
- B. File system
- C. Memory manager
- D. Process manager

Answer: B

Q19. What is the role of device drivers in an OS?

- A. To interface between the OS and hardware
- B. To manage user accounts and passwords
- C. To compile programs into machine code
- D. To execute user-level application programs

Answer: A

Q20. Linux is an example of which type of kernel?

- A. Monolithic kernel
- B. Nanokernel design
- C. Microkernel design
- D. Hybrid kernel

Answer: A

Q21. A process is:

- A. A single CPU register
- B. A program in execution
- C. A file held in memory
- D. A program stored on disk

Answer: B

Q22. Which of the following is NOT a state of a process?

- A. Compiling
- B. Ready
- C. Waiting
- D. Running

Answer: A

Q23. The Process Control Block (PCB) contains:

- A. Only the executable program code of the process
- B. Process state, program counter, CPU registers, etc
- C. Only the virtual and physical memory addresses
- D. Only the process name and its unique identifier

Answer: B

Q24. What system call creates a new process in UNIX/Linux?

- A. new()
- B. spawn()
- C. create()
- D. fork()

Answer: D

Q25. A context switch occurs when:

- A. A file is saved to secondary storage
- B. The user logs out of the session
- C. The CPU switches between processes
- D. A new program is installed on disk

Answer: C

Q26. An orphan process is:

- A. A process that has no memory allocated
- B. A child process whose parent has terminated
- C. A process that is waiting for I/O
- D. A process that is in the ready queue

Answer: B

Q27. A zombie process is:

- A. A process that is stuck inside an infinite loop
- B. A process running with very high execution priority
- C. A terminated process whose exit status is uncollected
- D. A process consuming excessive CPU cycles continuously

Answer: C

Q28. The program counter in a process stores:

- A. The address of the next instruction to execute
- B. The scheduling priority assigned to the process
- C. The total amount of memory currently in use
- D. The total count of programs currently running

Answer: A

Q29. Inter-process communication (IPC) allows:

- A. A process to access hardware directly
- B. A process to run faster on the CPU
- C. A process to compile source code files
- D. Processes to exchange data and synchronize

Answer: D

Q30. When a fork() system call is executed successfully, what does it return to the child process?

- A. Parent PID value
- B. Zero is returned
- C. Child PID value
- D. Negative one val

Answer: B

Q31. CPU scheduling decides:

- A. How much memory to allocate to each process
- B. Which ready process gets the CPU execution next
- C. Which files to open for the running processes
- D. Which device to use for the I/O operations

Answer: B

Q32. In First-Come, First-Served (FCFS) scheduling, processes are executed:

- A. In the order they arrive in the queue
- B. Based on their assigned priority level
- C. With the shortest job running first
- D. Randomly without any specific order

Answer: A

Q33. Turnaround time of a process is defined as:

- A. Total time from submission to completion of process
- B. Time the process spends waiting in the ready queue
- C. Time spent performing input and output operations
- D. Time the CPU is busy executing process instructions

Answer: A

Q34. Waiting time of a process is:

- A. The total time from arrival to completion
- B. The time spent waiting in the ready queue
- C. The time spent in I/O operations
- D. The time spent executing on the CPU

Answer: B

Q35. Round Robin scheduling uses:

- A. Random process allocation
- B. Longest job first execution
- C. A fixed time quantum per process
- D. Priority-based CPU allocation

Answer: C

Q36. Preemptive scheduling means:

- A. The OS can interrupt and reassign the CPU to another
- B. Processes schedule themselves without any OS control
- C. Only one process can ever run at a time on the CPU
- D. A process always runs until it completes execution

Answer: A

Q37. Non-preemptive scheduling means:

- A. The CPU can be taken away from a running process at any time
- B. Once a process starts, it runs until it finishes or blocks
- C. The scheduler never runs and the CPU sits idle indefinitely
- D. Multiple processes share the CPU at the same exact instant

Answer: B

Q38. Which scheduling algorithm gives the minimum average waiting time for a given set of processes?

- A. First-Come First-Served
- B. Priority Scheduling
- C. Shortest Job First SJF
- D. Round Robin scheduling

Answer: C

Q39. The dispatcher is responsible for:

- A. Selecting the next process to execute on CPU
- B. Managing main memory allocation for process
- C. Giving control of CPU to the selected process
- D. Handling file system operations for process

Answer: C

Q40. CPU utilization refers to:

- A. The number of processes completed per unit time
- B. The number of context switches performed per second
- C. The percentage of time the CPU is busy executing
- D. The total time a process waits in the ready queue

Answer: C

Q41. A thread is:

- A. A type of volatile memory block
- B. A lightweight execution unit in process
- C. A physical hardware component part
- D. A complete standalone process

Answer: B

Q42. Threads within the same process share all of the following EXCEPT:

- A. Data section
- B. Stack space
- C. Open files
- D. Code section

Answer: B

Q43. What is the main advantage of using threads over processes?

- A. Threads can run on different machines
- B. Threads are inherently more secure overall
- C. Thread creation and switching are faster
- D. Threads use significantly more memory

Answer: C

Q44. User-level threads are managed by:

- A. The file system layer
- B. A user-level thread library
- C. The hardware platform
- D. The kernel directly

Answer: B

Q45. Kernel-level threads are managed by:

- A. A web browser
- B. A compiler tool
- C. User applications
- D. The OS kernel

Answer: D

Q46. Multithreading benefits include all EXCEPT:

- A. Resource sharing across threads
- B. Responsiveness to user input
- C. Economy of creation and switching
- D. Guaranteed elimination of bugs

Answer: D

Q47. POSIX Pthreads is:

- A. A standard API for thread management
- B. A file system for multithreading
- C. A hardware component for threads
- D. An operating system distribution

Answer: A

Q48. In a single-threaded process, how many threads of control exist?

- A. One thread
- B. Many threads
- C. Two threads
- D. Zero threads

Answer: A

Q49. Which of the following is a benefit of multithreading in web server applications?

- A. It handles multiple client requests concurrently
- B. It reduces overall system security
- C. It simplifies the server source code logic
- D. It eliminates the need for any networking

Answer: A

Q50. A process can contain:

- A. One or more threads
- B. No threads at all
- C. Only two threads
- D. Only one thread

Answer: A

Q51. A race condition occurs when:

- A. Two processes have the same scheduling priority level
- B. Outcome depends on execution order of shared data access
- C. A process runs too fast for the system to handle
- D. Two processes run on the CPU simultaneously

Answer: B

Q52. The critical section is:

- A. The most important section of a running program
- B. Code segment accessing shared resources exclusively
- C. The first section executed when a program starts
- D. A section within the operating system kernel code

Answer: B

Q53. Mutual exclusion means:

- A. Processes must exclude each other from the system
- B. Two processes can be in critical sections at once
- C. No process is ever allowed to enter any section
- D. Only one process can be in its critical section

Answer: D

Q54. A mutex lock is:

- A. A hardware component on the motherboard
- B. A type of running process in the system
- C. A synchronization tool using acquire and release
- D. A type of physical memory module

Answer: C

Q55. A semaphore is:

- A. A type of hardware interrupt signal
- B. A CPU scheduling algorithm technique
- C. A type of boolean flag variable
- D. A variable accessed via wait and signal

Answer: D

Q56. A binary semaphore can have values:

- A. Only negative value range
- B. Any positive integer value
- C. Only zero and one values
- D. Any integer value at all

Answer: C

Q57. A counting semaphore can have values:

- A. Any non-negative integer value
- B. Only negative value ranges
- C. Only zero and one values
- D. Unrestricted integer domain

Answer: D

Q58. Busy waiting (spinlock) means:

- A. A process is waiting for I/O completion
- B. A process is terminated by the operating system
- C. A process continuously loops testing a condition
- D. A process is sleeping and waiting for a signal

Answer: C

Q59. What are the three requirements for a solution to the critical section problem?

- A. Atomicity, consistency, and isolation
- B. Fairness, priority, and preemption
- C. Mutual exclusion, progress, bounded waiting
- D. Speed, efficiency, and simplicity

Answer: C

Q60. The wait() operation on a semaphore is also known as:

- A. P operation (proberen)
- B. V operation (verhogen)
- C. Lock operation call
- D. Signal operation call

Answer: A

Q61. A deadlock occurs when:

- A. Processes wait indefinitely for resources held by each other
- B. A process enters an infinite loop consuming all CPU time
- C. The CPU is overloaded with too many running processes
- D. A process terminates unexpectedly due to an error

Answer: A

Q62. How many necessary conditions must hold simultaneously for a deadlock to occur?

- A. Three
- B. Four
- C. Two
- D. Five

Answer: B

Q63. Which of the following is NOT a necessary condition for deadlock?

- A. Mutual exclusion
- B. Hold and wait
- C. Circular wait
- D. Preemption

Answer: D

Q64. The 'hold and wait' condition means:

- A. A process waits without holding any resource
- B. A process holds no resources at all currently
- C. A process releases all resources before waiting
- D. A process holds resources while waiting for more

Answer: D

Q65. Circular wait means:

- A. Processes are scheduled in a circular fashion
- B. Each process waits for the next, forming a cycle
- C. Processes simply wait in a standard queue
- D. Only one single process is actively waiting

Answer: B

Q66. Deadlock prevention works by:

- A. Ignoring deadlocks entirely and hoping they pass
- B. Ensuring at least one necessary condition fails
- C. Recovering from deadlocks by terminating tasks
- D. Detecting deadlocks after they already occurred

Answer: B

Q67. A resource allocation graph is used to:

- A. Display the memory layout of processes
- B. Represent resource allocation and requests
- C. Show the CPU scheduling execution order
- D. Show the file system directory structure

Answer: B

Q68. If a resource allocation graph has no cycle, then:

- A. Deadlock definitely exists in system
- B. Deadlock may or may not exist here
- C. Deadlock definitely does not exist
- D. The resource allocation graph invalid

Answer: C

Q69. Deadlock avoidance requires:

- A. Advance info about resource requests
- B. Using only one resource at a time
- C. Killing processes periodically needed
- D. No additional information at all

Answer: A

Q70. The Banker's Algorithm is used for:

- A. Checking if granting request is safe
- B. Processing banking transactions safely
- C. Memory allocation for new processes
- D. CPU scheduling of running processes

Answer: A

Q71. Main memory (RAM) is:

- A. Volatile storage the CPU accesses directly
- B. A backup device for storing archives
- C. Non-volatile persistent storage on disk
- D. A type of secondary storage for backup

Answer: A

Q72. Logical address is:

- A. The network address assigned to a device
- B. An address generated by CPU for translation
- C. The physical location in hardware memory
- D. The address of data on the hard disk drive

Answer: B

Q73. Physical address is:

- A. An address generated by the CPU logic
- B. A virtual address used by programs
- C. An IP address on the local network
- D. The actual address in memory hardware

Answer: D

Q74. The Memory Management Unit (MMU) is responsible for:

- A. Managing I/O device controllers
- B. Scheduling processes on the CPU
- C. Translating logical to physical addresses
- D. Managing the file system on disk

Answer: C

Q75. Paging divides:

- A. The disk into sectors and track clusters
- B. Processes into variable-size logical segments
- C. Physical memory into frames, logical into pages
- D. The CPU into separate processing partitions

Answer: C

Q76. A page table stores:

- A. CPU register contents
- B. File names and paths
- C. Process priority values
- D. Page to frame mappings

Answer: D

Q77. External fragmentation occurs when:

- A. The disk is completely full and cannot store data
- B. The allocated pages are much too large for use
- C. Internal memory within blocks is wasted by padding
- D. Free memory is scattered, cannot satisfy requests

Answer: D

Q78. Internal fragmentation occurs when:

- A. No memory is available to allocate to processes
- B. The file system is corrupted and cannot allocate blocks
- C. Allocated block is larger than requested, wasting space
- D. Free memory is scattered in small pieces across RAM

Answer: C

Q79. Virtual memory allows:

- A. Only one process to run at any given time
- B. Programs larger than physical memory to execute
- C. Faster CPU execution by clock overclocking
- D. Direct access to underlying hardware devices

Answer: B

Q80. A page fault occurs when:

- A. A referenced page is not in physical memory
- B. A page has become corrupted in memory
- C. The page table is completely full of entries
- D. A page is too large to fit in any frame

Answer: A

Q81. A file is:

- A. A physical storage device like a hard drive
- B. A hardware component on the computer motherboard
- C. A type of process running in the operating system
- D. A named collection of data on secondary storage

Answer: D

Q82. A directory in a file system:

- A. Is the same as a disk partition or volume
- B. Contains file info like names and locations
- C. Is a file type that stores only text content
- D. Stores raw data directly on the disk surface

Answer: B

Q83. Which file access method reads records in order, one after another?

- A. Random access
- B. Indexed access
- C. Sequential access
- D. Direct access

Answer: C

Q84. The absolute path of a file starts from:

- A. The root directory
- B. The home directory
- C. The current directory
- D. The parent directory

Answer: A

Q85. File attributes typically include:

- A. Name, type, size, location, timestamps
- B. Only the file creation date stored
- C. Only the file size in bytes stored
- D. Only the file name string value

Answer: A

Q86. An inode in UNIX/Linux file systems stores:

- A. Metadata: permissions, size, pointers
- B. The actual file content data itself
- C. The file name string value
- D. The directory name for this entry

Answer: A

Q87. A hard link in UNIX:

- A. Creates a complete copy of the original file
- B. Links to a file located on another disk drive
- C. Creates a shortcut file with a different path
- D. Creates another entry pointing to same inode

Answer: D

Q88. A symbolic (soft) link is:

- A. A compressed version of original file
- B. A file containing path to another file
- C. A full copy of the original source file
- D. The same as a hard link to a file

Answer: B

Q89. File system mounting is the process of:

- A. Deleting a file system from a partition
- B. Compressing a file system for archiving
- C. Creating a new file system on disk
- D. Making a file system accessible at a point

Answer: D

Q90. FAT (File Allocation Table) file system uses:

- A. Inodes for each file entry
- B. A table of linked block lists
- C. Hash tables for fast lookup
- D. B-trees for indexing entries

Answer: B

Q91. Secondary storage refers to:

- A. Processor register files
- B. Non-volatile storage like disks
- C. Random access memory chips
- D. CPU cache memory modules

Answer: B

Q92. A hard disk drive (HDD) stores data on:

- A. Flash memory components
- B. Semiconductor memory chips
- C. Rotating magnetic platters
- D. Optical disc technology

Answer: C

Q93. Seek time in a disk drive is:

- A. Time for the disk to spin to correct sector
- B. Time to transfer requested data from disk
- C. Time for the head to move to correct track
- D. Time to process and queue the I/O request

Answer: C

Q94. Rotational latency is:

- A. Time to start the disk motor from stopped
- B. Time for desired sector to rotate under head
- C. Time for the head to move to correct track
- D. Time to format a disk sector with structure

Answer: B

Q95. A Solid State Drive (SSD) stores data using:

- A. Magnetic tape reels
- B. Magnetic platters
- C. Flash memory chips
- D. Optical technology

Answer: C

Q96. FCFS disk scheduling processes disk I/O requests:

- A. Based on closest track first
- B. In the order they arrive
- C. In reverse arrival order
- D. Based on request priority

Answer: B

Q97. The SSTF (Shortest Seek Time First) disk scheduling algorithm:

- A. Services the closest request to head
- B. Always moves the head toward outer
- C. Services the largest request first
- D. Services all requests in arrival order

Answer: A

Q98. Disk formatting prepares a disk for use by:

- A. Deleting all viruses from the disk
- B. Creating file system structure on disk
- C. Increasing the disk rotational speed
- D. Adding more storage space to a disk

Answer: B

Q99. A disk partition is:

- A. A physical disk drive unit in chassis
- B. A type of data file stored on the disk
- C. A backup copy of the entire disk drive
- D. An independent section with own file system

Answer: D

Q100. Disk access time is the sum of:

- A. Only the seek time component alone
- B. Only the data transfer time component
- C. Only the rotational latency component
- D. Seek time, rotational latency, transfer time

Answer: D

Q101. I/O devices communicate with the CPU through:

- A. The file system interface
- B. Direct wiring to CPU
- C. Shared memory regions only
- D. Device controllers and buses

Answer: D

Q102. Polling in I/O means:

- A. CPU repeatedly checks device status register
- B. Voting on which device to use for the task
- C. Automatically detecting new devices on the bus
- D. Prioritizing I/O requests based on their urgency

Answer: A

Q103. Interrupt-driven I/O means:

- A. I/O operations are interrupted frequently
- B. The CPU is constantly being interrupted
- C. The CPU stops all I/O until processing done
- D. The device signals CPU when I/O is complete

Answer: D

Q104. DMA (Direct Memory Access) allows:

- A. Direct access to the motherboard components
- B. Devices to transfer data to memory without CPU
- C. The CPU to access main memory much faster
- D. Memory to access the CPU directly for data

Answer: B

Q105. A device driver is:

- A. Software providing OS-to-hardware interface
- B. A type of cable connecting to the bus
- C. A person who physically operates devices
- D. A diagnostic tool for testing hardware

Answer: A

Q106. Block devices transfer data in:

- A. Variable packets
- B. Single characters
- C. Continuous streams
- D. Fixed-size blocks

Answer: D

Q107. Character devices transfer data:

- A. One byte at a time in a stream
- B. In fixed-size data blocks
- C. In variable-size network packets
- D. In fixed-size memory page frames

Answer: A

Q108. A bus in computer architecture is:

- A. A type of vehicle for transport use
- B. A type of volatile memory used in computers
- C. A processing unit that executes instructions
- D. A shared pathway transferring data between parts

Answer: D

Q109. Buffering in I/O systems means:

- A. Encrypting I/O data for security purposes
- B. Compressing I/O data for disk storage
- C. Temporarily storing data during transfer
- D. Slowing down I/O operations intentionally

Answer: C

Q110. Spooling is commonly used for:

- A. CPU scheduling for processes
- B. Network routing of data packets
- C. Memory management of processes
- D. Managing print jobs via spooling

Answer: D

Q111. Protection in operating systems refers to:

- A. Controlling process and user access to resources
- B. Antivirus software scanning for malware threats
- C. Firewall configuration for network protection
- D. Physical security of the computer hardware

Answer: A

Q112. Security in operating systems deals with:

- A. Only network security from remote attacks
- B. Defending against internal and external attacks
- C. Only internal threats from local users
- D. Only physical security of the server room

Answer: B

Q113. Authentication is the process of:

- A. Verifying identity of a user
- B. Encrypting data for security
- C. Authorizing resource access
- D. Backing up critical data

Answer: A

Q114. Authorization determines:

- A. When to back up data for disaster recovery
- B. How to encrypt data for secure communication
- C. Who the user is based on their credentials
- D. What resources a verified user may access

Answer: D

Q115. The principle of least privilege states that:

- A. All users should have full administrator access rights
- B. The least privileged user should have no access at all
- C. Users get only the minimum access needed for tasks
- D. All privileges should be equal for all system users

Answer: C

Q116. A virus in computer security is:

- A. Self-replicating malicious program on host
- B. A network communication protocol standard
- C. A hardware malfunction in the CPU unit
- D. A type of data encryption algorithm used

Answer: A

Q117. A worm differs from a virus in that:

- A. A worm only affects the physical hardware parts
- B. A worm replicates across networks without host
- C. A worm is less harmful than a virus overall
- D. A worm cannot spread to other network systems

Answer: B

Q118. A Trojan horse in computer security is:

- A. A system backup utility tool for data
- B. A type of network firewall tool used
- C. A type of hardware peripheral device
- D. Malicious program disguised as legitimate

Answer: D

Q119. Encryption is the process of:

- A. Converting plaintext to ciphertext with key
- B. Backing up data to a remote server
- C. Deleting data permanently from disk
- D. Compressing data for efficient storage

Answer: A

Q120. A firewall is:

- A. A system monitoring and controlling network traffic
- B. A physical wall in the server room for safety
- C. A type of computer virus that spreads via network
- D. A backup system for disaster recovery procedures

Answer: A

Q121. Which component loads the operating system into memory during startup?

- A. The application manager
- B. The device controller
- C. The bootstrap loader
- D. The system compiler

Answer: C

Q122. What does a distributed operating system manage?

- A. Multiple networked computers together
- B. A single standalone desktop machine
- C. Only mobile phone applications here
- D. Exclusively embedded device controllers

Answer: A

Q123. Which of the following best describes firmware?

- A. Programs permanently stored in hardware
- B. Software stored in volatile RAM chips
- C. Temporary data written to disk drives
- D. User applications installed externally

Answer: A

Q124. What is the primary purpose of an interrupt in a computer system?

- A. To delete all currently running tasks
- B. To signal the CPU about an event
- C. To format all storage device media
- D. To permanently halt all processors

Answer: B

Q125. Which type of OS is designed for devices with very limited resources?

- A. Embedded system OS
- B. Desktop general OS
- C. Mainframe batch OS
- D. Enterprise server OS

Answer: A

Q126. What does GUI stand for in the context of operating systems?

- A. Graphical User Interface
- B. General Unified Interface
- C. Global Usage Indicator
- D. Generated Utility Index

Answer: A

Q127. Which OS function manages the allocation and deallocation of main memory?

- A. Network management
- B. User authentication
- C. Disk formatting
- D. Memory management

Answer: D

Q128. What is spooling in an operating system context?

- A. Buffering data for slow output devices
- B. Compressing files to save disk storage
- C. Defragmenting the hard disk sectors
- D. Encrypting data before transmitting out

Answer: A

Q129. Which of the following is NOT a type of operating system?

- A. Real-time operating system
- B. Network operating system
- C. Distributed operating system
- D. Compiled operating system

Answer: D

Q130. What is the role of device drivers in an operating system?

- A. They enable OS-hardware communication
- B. They translate high-level languages
- C. They manage user account passwords
- D. They compile user application code

Answer: A

Q131. What is a monolithic kernel operating system?

- A. An OS where all services run in kernel space
- B. An OS that only runs a single application task
- C. An OS designed for mobile devices exclusively
- D. An OS with no kernel component present at all

Answer: A

Q132. Which of the following is an example of a system call category?

- A. Process control
- B. Web browsing
- C. Graphics rendering
- D. Text formatting

Answer: A

Q133. What is the purpose of an API in an operating system?

- A. It manages physical hardware components alone
- B. It replaces the kernel entirely in modern OS
- C. It provides functions for programmers to use
- D. It handles only network communication protocol

Answer: C

Q134. What does the command interpreter do in an operating system?

- A. It controls hardware interrupt processing now
- B. It compiles all programs into machine code
- C. It manages virtual memory page tables only
- D. It reads and executes user commands directly

Answer: D

Q135. Which layer is at the bottom in a layered operating system design?

- A. File system layer
- B. Hardware layer
- C. Application layer
- D. User interface layer

Answer: B

Q136. What is a system program in the context of an operating system?

- A. A utility program bundled with the OS itself
- B. A virus that attacks the operating system now
- C. A program that only runs during system boot
- D. A diagnostic tool built into the CPU hardware

Answer: A

Q137. What type of system call is used to create a new process?

- A. Communication call
- B. File management call
- C. Device management call
- D. Process control call

Answer: D

Q138. What is the function of a linker in program execution?

- A. It translates source code to assembly code
- B. It handles input and output device operation
- C. It manages memory allocation during runtime
- D. It combines object modules into executable

Answer: D

Q139. Which of the following is a communication system call?

- A. open() call
- B. exit() call
- C. pipe() call
- D. fork() call

Answer: C

Q140. What is the role of a loader in program execution?

- A. It manages the network connections system
- B. It designs the user interface for programs
- C. It writes source code for the programmer
- D. It loads executable files into main memory

Answer: D

Q141. What is a process in an operating system?

- A. A program in execution in main memory
- B. A type of network protocol standard
- C. A hardware component inside the CPU
- D. A file stored on the hard disk drive

Answer: A

Q142. Which of the following is a valid process state?

- A. Formatting
- B. Installing
- C. Waiting
- D. Compiling

Answer: C

Q143. What does the Process Control Block contain?

- A. All user passwords and credentials
- B. Only the program source code text
- C. Information about a specific process
- D. The entire operating system kernel

Answer: C

Q144. What system call creates a new process in Unix?

- A. exec()
- B. fork()
- C. kill()
- D. wait()

Answer: B

Q145. What is a parent process in operating system terminology?

- A. The first process created during boot only
- B. A process with the highest priority in queue
- C. A process that created one or more children
- D. A process managing all hardware device drivers

Answer: C

Q146. What happens when a process moves from running to waiting state?

- A. The process begins on a different CPU core
- B. The process is permanently terminated now
- C. The process waits for an event like I/O
- D. The process moves back to the ready queue

Answer: C

Q147. What is the ready queue in process management?

- A. Processes that are ready and waiting for CPU
- B. A list of processes waiting to load from disk
- C. Blocked processes waiting for user keyboard input
- D. A queue of completed processes to be deleted

Answer: A

Q148. What is the purpose of the process scheduler?

- A. To handle network packet routing operations
- B. To manage file system directory structures now
- C. To select a process from ready queue for CPU
- D. To compile programs into executable files

Answer: C

Q149. What is process termination in operating systems?

- A. Pausing a process temporarily for later
- B. Ending a process and releasing its resources
- C. Changing the priority of a running process
- D. Moving a process to another computer node

Answer: B

Q150. What is an orphan process in Unix/Linux systems?

- A. A process consuming no memory or CPU at all
- B. A process whose parent has terminated first
- C. A process running only during system shutdown
- D. A process that has no executable code to run

Answer: B

Q151. What is CPU scheduling in an operating system?

- A. Overclocking the CPU for better processing performance
- B. Installing new processors into the computer motherboard
- C. Deciding which process gets the CPU next for execution
- D. Measuring the temperature of the CPU chip continuously

Answer: C

Q152. What does FCFS stand for in CPU scheduling?

- A. First Cache First Store
- B. Fast CPU Flexible Scheduling
- C. Fully Controlled File System
- D. First Come First Served

Answer: D

Q153. What is turnaround time in CPU scheduling?

- A. Time to switch between two processes
- B. Total time from submission to completion
- C. The speed at which the CPU clock runs
- D. The time to restart the computer

Answer: B

Q154. What is waiting time in CPU scheduling?

- A. Time needed to compile a program from source
- B. Time required to boot the operating system
- C. Time a process spends in the ready queue
- D. Time taken to save a file to the disk drive

Answer: C

Q155. What is the time quantum in Round Robin scheduling?

- A. The time required for a context switch now
- B. The clock speed of the processor chip now
- C. The total time to complete all processes
- D. A fixed time slice given to each process

Answer: D

Q156. Which scheduling algorithm selects the shortest job first?

- A. First Come First Served
- B. Round Robin scheduling
- C. Shortest Job First
- D. Priority scheduling only

Answer: C

Q157. What is throughput in CPU scheduling metrics?

- A. The speed of data transfer to the disk drive
- B. Number of processes completed per time unit
- C. The temperature of CPU during operation
- D. The size of the ready queue at a given moment

Answer: B

Q158. What is a preemptive scheduling algorithm?

- A. One where OS can take CPU from a process
- B. One that only works with a single process
- C. One that requires manual user intervention
- D. One that never interrupts a running process

Answer: A

Q159. What is a CPU burst in process execution?

- A. A period of continuous CPU computation time
- B. A malfunction causing the CPU to overheat now
- C. A type of interrupt from external hardware now
- D. A period when the CPU is powered off safely

Answer: A

Q160. In priority scheduling, which process gets the CPU first?

- A. The process with lowest priority value
- B. The process with highest priority value
- C. The process arriving most recently now
- D. The process with longest burst time

Answer: B

Q161. What is a thread in operating system terminology?

- A. A network connection between two machines
- B. A type of file stored on the hard disk
- C. A lightweight execution unit in a process
- D. A separate physical CPU core in system

Answer: C

Q162. What do threads within the same process share?

- A. Nothing at all between them
- B. Individual stack memory only
- C. Code, data, and file resources
- D. Only the process identifier

Answer: C

Q163. What is multithreading in operating systems?

- A. Running multiple threads within a single process
- B. Running multiple operating systems simultaneously
- C. Using multiple monitors for display output only
- D. Connecting multiple keyboards to one computer

Answer: A

Q164. Which of the following is a benefit of using threads?

- A. Threads prevent all forms of parallel execution
- B. Thread creation is faster than process creation
- C. Threads consume more resources than processes
- D. Threads restrict programs to a single CPU core

Answer: B

Q165. What is a user-level thread managed by?

- A. The operating system kernel directly
- B. A user-level thread library only
- C. The network interface card alone
- D. The hardware CPU scheduler chip

Answer: B

Q166. What is a kernel-level thread?

- A. A thread that manages hardware device interrupts
- B. A thread that only runs during system boot time
- C. A thread managed and scheduled by the kernel
- D. A thread that exists only in embedded OS systems

Answer: C

Q167. What does each thread have that is unique to itself?

- A. Its own copy of all open file handles now
- B. Its own operating system kernel instance
- C. Its own stack and program counter value
- D. Its own copy of entire process memory

Answer: C

Q168. What is the main advantage of multithreading in a web server?

- A. It prevents any client from accessing the server
- B. It reduces the server network bandwidth usage
- C. It eliminates the need for any network protocol
- D. It handles multiple client requests concurrently

Answer: D

Q169. What is the Pthreads API?

- A. A POSIX standard for thread management
- B. A network protocol for file transferring
- C. A graphics library for rendering images
- D. A database query language for data access

Answer: A

Q170. What happens when one thread in a process calls exit()?

- A. Other threads continue running without any impact
- B. Only that specific thread is terminated safely
- C. The entire process and all its threads terminate
- D. The thread is moved to a different process now

Answer: C

Q171. What is a race condition in concurrent programming?

- A. A technique for sorting data in parallel
- B. A way to synchronize clocks between PCs
- C. When output depends on timing of threads
- D. A method to speed up program execution

Answer: C

Q172. What is mutual exclusion in process synchronization?

- A. Allowing multiple processes to access resources
- B. Preventing any process from executing on CPU
- C. Deleting shared resources to prevent any access
- D. Ensuring only one process in critical section now

Answer: D

Q173. What is a critical section in concurrent programming?

- A. The first section of code executed by program
- B. The most important function in program code
- C. A section of code needing no synchronization
- D. A code segment accessing shared resources here

Answer: D

Q174. What is a semaphore in operating systems?

- A. A synchronization variable with atomic ops
- B. A memory allocation technique for processes
- C. A scheduling algorithm for real-time tasks
- D. A type of hardware interrupt signal line

Answer: A

Q175. What is the purpose of a mutex lock?

- A. To handle network packet routing faster
- B. To speed up single-thread execution
- C. To ensure mutual exclusion in critical section
- D. To manage memory allocation for processes

Answer: C

Q176. What does the wait() operation do on a semaphore?

- A. It creates a new semaphore with given value
- B. It decrements value and blocks if negative
- C. It increments the semaphore value by one
- D. It destroys the semaphore variable for good

Answer: B

Q177. What does signal() do on a semaphore?

- A. It increments value and may wake a process
- B. It deletes the semaphore from the system
- C. It resets semaphore to its initial value now
- D. It decrements semaphore value by one unit

Answer: A

Q178. What is a binary semaphore?

- A. A semaphore requiring two processes only
- B. A semaphore restricted to zero or one
- C. A semaphore holding any integer value
- D. A semaphore for binary search algorithms

Answer: B

Q179. What is busy waiting in the context of synchronization?

- A. A thread sleeps until explicitly woken up
- B. A thread exits when a resource is locked
- C. A thread continuously checks a condition
- D. A thread creates more threads to help out

Answer: C

Q180. What is a counting semaphore used for?

- A. Measuring CPU clock speed in gigahertz
- B. Counting total processes in the OS system
- C. Controlling access to finite resource instances
- D. Counting the number of files on the disk

Answer: C

Q181. What is a deadlock in an operating system?

- A. When processes wait for each other forever
- B. When memory is fully available to all tasks
- C. When the CPU overheats and shuts down
- D. When a program runs faster than expected

Answer: A

Q182. How many conditions must hold simultaneously for deadlock?

- A. One
- B. Four
- C. Two
- D. Three

Answer: B

Q183. What is the mutual exclusion condition for deadlock?

- A. Processes can preempt resources from other ones
- B. Resources can be shared by multiple processes
- C. All resources are always available to every process
- D. At least one resource must be held non-sharably

Answer: D

Q184. What is the hold and wait condition for deadlock?

- A. Processes never hold more than one resource at a time
- B. Processes wait without holding any resources at all
- C. Processes hold resources while waiting for others
- D. Processes release resources before requesting new ones

Answer: C

Q185. What is a resource allocation graph used for?

- A. Designing the layout of network infrastructure
- B. Measuring CPU utilization over time periods
- C. Drawing flowcharts for program logic design
- D. Visualizing allocation and detecting deadlock

Answer: D

Q186. What is the circular wait condition for deadlock?

- A. Processes wait for resources in random order
- B. A circular chain where each waits for the next
- C. Processes wait in a linear queue for resources
- D. All processes wait for the same single resource

Answer: B

Q187. What is the no preemption condition for deadlock?

- A. Only the OS kernel can hold resources permanently
- B. Resources cannot be preempted from holders now
- C. Resources can be forcibly taken from processes
- D. All resources are preemptible at any time freely

Answer: B

Q188. What is deadlock prevention as a strategy?

- A. Recovering by terminating all running processes
- B. Ignoring deadlocks and hoping they never occur
- C. Detecting deadlocks after they already occurred
- D. Ensuring at least one deadlock condition fails

Answer: D

Q189. What is starvation in resource allocation?

- A. A process runs out of memory and crashes fast
- B. A process is denied needed resources indefinitely
- C. A process completes too quickly and wastes resources
- D. A process gets all resources it needs quickly

Answer: B

Q190. What is a safe state in deadlock avoidance?

- A. A state where resources can be allocated avoiding deadlock
- B. A state where all resources are fully allocated already
- C. A state where the system has no processes running
- D. A state where CPU temperature is within safe limits

Answer: A

Q191. What is the main purpose of memory management?

- A. To permanently store data like a hard drive
- B. To increase the physical size of RAM chips
- C. To manage network bandwidth between hosts
- D. To allocate and track memory for processes

Answer: D

Q192. What is virtual memory in an operating system?

- A. A type of cache memory inside the CPU chip
- B. A technique giving illusion of larger memory
- C. Physical RAM installed in the computer
- D. A backup copy of files stored on the disk

Answer: B

Q193. What is a page in the context of virtual memory?

- A. A type of hardware interrupt signal to CPU
- B. A section of the operating system kernel
- C. A unit of data stored on the hard disk now
- D. A fixed-size block of virtual memory space

Answer: D

Q194. What is a page table used for?

- A. Displaying a table of contents for docs
- B. Storing user passwords and credentials now
- C. Mapping virtual addresses to physical ones
- D. Scheduling CPU time for running processes

Answer: C

Q195. What is a page fault in virtual memory systems?

- A. A syntax error in the running program code
- B. An access to a page not in main memory now
- C. A hardware malfunction in memory module
- D. A failure of CPU to execute an instruction

Answer: B

Q196. What is a frame in physical memory?

- A. A fixed-size block of physical RAM memory
- B. A unit of data transmitted over a network
- C. A type of CPU register storing addresses
- D. A decorative border around monitor screen

Answer: A

Q197. What is swapping in memory management?

- A. Exchanging CPU cores between processes
- B. Replacing old hardware with new hardware
- C. Switching between different user accounts
- D. Moving process between memory and disk

Answer: D

Q198. What is internal fragmentation in memory?

- A. Memory corrupted due to hardware failure
- B. File fragments scattered across the disk
- C. Free memory between allocated blocks here
- D. Wasted space inside an allocated partition

Answer: D

Q199. What is external fragmentation in memory?

- A. A process using more than available memory
- B. Memory corruption from external hardware
- C. Free memory scattered in small pieces now
- D. Memory allocated to already exited processes

Answer: C

Q200. What is the logical address space of a process?

- A. Address of the process control block here
- B. Set of all addresses generated by process
- C. Total amount of RAM installed in computer
- D. Physical location of process on the disk

Answer: B

Q201. What is a file in operating system terminology?

- A. A network protocol for data transfer
- B. A physical storage device like a disk
- C. A type of CPU instruction for processing
- D. A named collection of related data items

Answer: D

Q202. What is a directory in a file system?

- A. A hardware device for reading media
- B. A physical partition on the hard disk
- C. A structure containing info about files
- D. A file type that cannot be modified now

Answer: C

Q203. What is a file extension used for?

- A. To indicate the type or format of a file
- B. To increase the file size beyond limits
- C. To encrypt the file contents securely now
- D. To compress files for saving disk space

Answer: A

Q204. What is the root directory in a file system?

- A. Where all temporary files are stored
- B. A directory only for administrators here
- C. A directory storing the OS kernel files
- D. The topmost directory in the hierarchy

Answer: D

Q205. What is a file's access permission used for?

- A. Determining the file size on the disk
- B. Setting the creation date of the entry
- C. Controlling who can read, write, execute
- D. Choosing which app opens the file type

Answer: C

Q206. What is sequential file access?

- A. Writing files in reverse order from the end
- B. Accessing files from multiple disks at once
- C. Accessing any record directly by position
- D. Reading records one after another in order

Answer: D

Q207. What is the purpose of file system mounting?

- A. Creating a backup copy of the file system
- B. Making a file system accessible at a location
- C. Physically installing new hard drive hardware
- D. Removing a file system from the OS entirely

Answer: B

Q208. What does the open() system call do for files?

- A. It permanently deletes a file from system
- B. It creates a brand new empty file on disk
- C. It prepares a file for reading or writing
- D. It changes the permissions of a file now

Answer: C

Q209. What is an absolute path in a file system?

- A. A path containing only the file name itself
- B. A path relative to current working directory
- C. A complete path starting from root directory
- D. A path that points to a deleted file now

Answer: C

Q210. What is a symbolic link in a file system?

- A. A file containing a path to another file
- B. A physical copy of a file in another place
- C. A special file storing encrypted disk data
- D. A temporary file created during system boot

Answer: A

Q211. What is secondary storage in a computer system?

- A. Non-volatile storage for permanent data
- B. The main memory used by running procs
- C. The CPU cache memory for fast access
- D. Registers inside the CPU for computation

Answer: A

Q212. What is a hard disk drive composed of?

- A. Optical discs using laser beams to read the data
- B. Solid-state flash memory chips sealed in case
- C. Spinning magnetic platters with read-write heads
- D. Volatile memory modules losing data when off now

Answer: C

Q213. What is disk seek time?

- A. Time to transfer data from disk to memory
- B. Time to format the entire disk from scratch
- C. Time to move head to the correct track now
- D. Time for platter to complete one rotation

Answer: C

Q214. What is a solid-state drive (SSD)?

- A. A mechanical drive with spinning platters
- B. A volatile device requiring constant power now
- C. Storage using flash memory, no moving parts
- D. A type of optical disc using laser for data

Answer: C

Q215. What is disk rotational latency?

- A. Total time to read an entire disk platter now
- B. Time for head to move to the correct track
- C. Time for desired sector to rotate under head
- D. Time required to start spinning disk platters

Answer: C

Q216. What is a disk partition?

- A. A hardware device connecting disks to CPU
- B. A logical division of disk into sections
- C. A type of disk error corrupting stored data
- D. A physical break in disk platter surface

Answer: B

Q217. What is the transfer rate of a disk?

- A. Speed at which disk platters rotate per minute
- B. Time taken to partition the disk into sections
- C. Rate of data flow between disk and computer now
- D. Rate at which disk head moves across all tracks

Answer: C

Q218. What is a disk track on a magnetic hard drive?

- A. A type of error correction code on the disk
- B. A line connecting disk to the motherboard now
- C. A physical groove cut into disk for head path
- D. A concentric circle on disk platter surface

Answer: D

Q219. What is a sector on a hard disk drive?

- A. Total storage capacity of one disk platter now
- B. A complete circular track on the platter now
- C. Smallest addressable unit of storage on disk
- D. Hardware component controlling the read head

Answer: C

Q220. What is RAID in secondary storage?

- A. A method to encrypt data on single disk
- B. A type of RAM for caching disk operations
- C. A protocol for transferring data on network
- D. Combining multiple disks for redundancy

Answer: D

Q221. What is an I/O device in a computer system?

- A. A device that only stores the OS kernel
- B. A device generating electrical power supply
- C. A device used for input, output, or both
- D. A device that only processes internal data

Answer: C

Q222. What is a device driver in an operating system?

- A. Software controlling a specific hardware device
- B. A person who operates hardware devices
- C. A physical cable connecting devices to the PC
- D. A tool for testing hardware functionality now

Answer: A

Q223. What is programmed I/O for device communication?

- A. Data transfers directly between device and memory
- B. OS ignores device until a user requests the data
- C. CPU continuously checks device status for data
- D. Device notifies CPU when it completes operation

Answer: C

Q224. What is interrupt-driven I/O?

- A. CPU continuously polls device for data status
- B. OS disables all interrupts during I/O operations
- C. Data transfers without any CPU involvement here
- D. Device signals CPU via interrupt when it is ready

Answer: D

Q225. What is Direct Memory Access used for?

- A. Transferring data between I/O devices and memory
- B. Directly accessing CPU registers from a program
- C. Providing direct access to the OS kernel code now
- D. Managing memory allocation for running processes

Answer: A

Q226. What is a block device in I/O classification?

- A. A device transferring data one character at a time
- B. A device that blocks all other devices from running
- C. A device transferring data in fixed-size blocks now
- D. A device working only when the system is blocked

Answer: C

Q227. What is a character device in I/O classification?

- A. A device storing character encoding tables on disk
- B. A device displaying only text characters on screen
- C. A device transferring data one character at a time
- D. A device used for character recognition in documents

Answer: C

Q228. What is the purpose of an I/O port?

- A. An address for CPU to communicate with devices
- B. A physical hole for ventilation of the case
- C. A port used only for internet connections now
- D. A type of software app for managing printers

Answer: A

Q229. What is buffering in I/O systems?

- A. Encrypting data before sending to output devices
- B. Temporarily storing data in transit between devices
- C. Permanently storing data on the hard disk drive now
- D. Deleting data no longer needed by any process here

Answer: B

Q230. What is the I/O request lifecycle?

- A. Time from device manufacture to its disposal
- B. Lifespan of cable connecting device to computer
- C. Sequence from application request to I/O completion
- D. Period when I/O device is powered on and running

Answer: C

Q231. What is the difference between protection and security?

- A. Protection controls internal; security guards external
- B. Neither is relevant to modern operating systems
- C. Security controls internal; protection guards external
- D. Protection and security mean the same thing

Answer: A

Q232. What is authentication in OS security?

- A. Installing antivirus software on the system
- B. Compressing data to save storage on disk
- C. Verifying the identity of a user or process
- D. Encrypting all files stored on the disk

Answer: C

Q233. What is an access control list in operating systems?

- A. A list specifying user permissions for objects
- B. A list of all network connections currently
- C. A list of all hardware devices attached now
- D. A list of all software installed on PC

Answer: A

Q234. What is a password used for in OS security?

- A. Encrypting the entire hard disk contents
- B. Compressing files to save storage on disk
- C. Increasing the processing speed of the CPU
- D. Authenticating user identity before access

Answer: D

Q235. What is malware in computer security?

- A. A type of hardware failure in the computer
- B. Malicious software designed to harm systems
- C. Software improving computer performance
- D. A legitimate tool for administration tasks

Answer: B

Q236. What is encryption used for in OS security?

- A. Converting data to unreadable format for privacy
- B. Deleting sensitive files permanently from the disk
- C. Compressing files to reduce their storage size now
- D. Increasing the speed of data transfer on network

Answer: A

Q237. What is a firewall in computer security?

- A. A tool for formatting hard drives securely now
- B. A device supplying emergency power to the PC
- C. A physical wall preventing computer overheating
- D. A system filtering network traffic based on rules

Answer: D

Q238. What is a computer virus?

- A. A tool for legitimate system maintenance tasks
- B. A type of operating system for mobile phones
- C. A self-replicating program attaching to files
- D. A hardware malfunction from overheating chips

Answer: C

Q239. What is the principle of least privilege?

- A. Granting users only minimum permissions needed
- B. Assigning same permissions to every system user
- C. Giving every user full administrator access
- D. Preventing all users from accessing the system

Answer: A

Q240. What is a Trojan horse in computer security?

- A. A security tool protecting against all viruses
- B. A hardware device used for network security
- C. A backup system storing copies of vital files
- D. Malware disguised as legitimate software here

Answer: D

Q241. Which of the following is a primary goal of an operating system?

- A. Convenience for the user
- B. Designing hardware circuits
- C. Writing application software
- D. Manufacturing peripheral devices

Answer: A

Q242. What is a batch operating system?

- A. An OS that processes jobs in groups without user interaction
- B. An OS that allows real-time user input during execution
- C. An OS designed only for mobile devices
- D. An OS that runs a single program at a time with user control

Answer: A

Q243. Which of the following best describes a time-sharing operating system?

- A. An OS that allocates dedicated hardware to each user
- B. An OS that allows multiple users to share CPU time interactively
- C. An OS that processes only one task at a time
- D. An OS used exclusively for scientific calculations

Answer: B

Q244. What does the term 'kernel mode' mean in an operating system?

- A. A privileged execution mode with full hardware access
- B. A mode where only user applications can run
- C. A power-saving mode that reduces CPU speed
- D. A mode used exclusively during system shutdown

Answer: A

Q245. Which component of a computer system directly executes instructions?

- A. Central Processing Unit (CPU)
- B. Hard disk drive
- C. Network interface card
- D. Power supply unit

Answer: A

Q246. What is multiprogramming in operating systems?

- A. Running multiple programs in memory to maximize CPU utilization
- B. Running the same program on multiple computers
- C. Writing programs in multiple programming languages
- D. Using multiple monitors for display output

Answer: A

Q247. What is the function of an operating system's resource manager?

- A. Allocating and managing hardware and software resources
- B. Designing new hardware components
- C. Writing user application programs
- D. Connecting to the internet

Answer: A

Q248. Which of the following is a network operating system?

- A. Windows Server
- B. MS-DOS
- C. Android
- D. Embedded Linux

Answer: A

Q249. What is the purpose of a command-line interface in an operating system?

- A. To allow users to interact with the OS by typing text commands
- B. To display graphical windows and icons
- C. To manage the file allocation table
- D. To compile source code into machine language

Answer: A

Q250. What is an embedded operating system?

- A. An OS designed for specialized devices with limited resources
- B. An OS used only on desktop computers
- C. An OS that requires a graphical user interface
- D. An OS that runs only in cloud environments

Answer: A

Q251. What is a system call used for in an operating system?

- A. Requesting services from the OS kernel
- B. Directly accessing hardware without the OS
- C. Compiling programs into machine code
- D. Formatting a hard disk drive

Answer: A

Q252. What is a shell in an operating system?

- A. A command interpreter that accepts and executes user commands
- B. The innermost part of the operating system kernel
- C. A hardware component on the motherboard
- D. A type of system memory

Answer: A

Q253. What type of system call is 'read()' in Unix?

- A. File management system call
- B. Process control system call
- C. Device management system call
- D. Communication system call

Answer: A

Q254. In a layered OS design, which layer interacts directly with the hardware?

- A. The lowest layer (Layer 0)
- B. The highest layer (user interface)
- C. The middle layer
- D. All layers equally

Answer: A

Q255. What is a device management system call used for?

- A. Requesting and releasing hardware devices
- B. Creating new files on disk
- C. Terminating running processes
- D. Sending messages between computers

Answer: A

Q256. What is the purpose of a boot loader?

- A. To load the operating system kernel into memory during startup
- B. To install new application software
- C. To manage user login credentials
- D. To defragment the hard disk

Answer: A

Q257. Which of the following is a process control system call?

- A. fork()
- B. open()
- C. read()
- D. chmod()

Answer: A

Q258. What is an operating system kernel?

- A. The core component of the OS that is always resident in memory
- B. A user-level application for managing files
- C. An external hardware device controller
- D. A network communication protocol

Answer: A

Q259. What is a system utility program?

- A. A program that provides common system functions like file management
- B. The core kernel of the operating system
- C. A hardware diagnostic chip on the motherboard
- D. An application for browsing the internet

Answer: A

Q260. What does the 'write()' system call do?

- A. Writes data to a file or output device
- B. Creates a new directory
- C. Deletes a file from disk
- D. Changes file permissions

Answer: A

Q261. What is a process state in operating systems?

- A. The current activity or condition of a process during its lifecycle
- B. The physical location of the process on disk
- C. The programming language used to write the process
- D. The amount of memory installed in the system

Answer: A

Q262. What happens when a running process requests an I/O operation?

- A. It moves from the running state to the waiting state
- B. It moves from the running state to the terminated state
- C. It continues executing without any state change
- D. It is permanently removed from memory

Answer: A

Q263. What information does the process ID (PID) provide?

- A. A unique numerical identifier assigned to each process by the OS
- B. The memory address where the process is stored
- C. The priority level of the process
- D. The name of the user who started the process

Answer: A

Q264. What is a child process?

- A. A process created by another process using a system call like fork()
- B. A process that has the highest priority in the system
- C. A process that manages hardware devices
- D. A process that is waiting for user input

Answer: A

Q265. What is the 'new' state of a process?

- A. The process is being created and has not yet been admitted to the ready queue
- B. The process is currently executing on the CPU
- C. The process is waiting for an I/O event to complete
- D. The process has finished execution

Answer: A

Q266. What is the 'terminated' state of a process?

- A. The process has finished execution and is being cleaned up by the OS
- B. The process is actively running on the CPU
- C. The process is waiting in the ready queue
- D. The process is blocked on an I/O request

Answer: A

Q267. What does the wait() system call do in Unix?

- A. Causes the parent process to wait until a child process terminates
- B. Pauses the CPU clock for a specified duration
- C. Waits for user keyboard input
- D. Delays the boot sequence of the operating system

Answer: A

Q268. What is a process table in an operating system?

- A. A data structure that stores information about all active processes
- B. A physical table of hardware components in the computer
- C. A list of all installed software applications
- D. A directory of files stored on the hard disk

Answer: A

Q269. What is the ready state of a process?

- A. The process is loaded in memory and waiting for CPU allocation
- B. The process is currently executing instructions
- C. The process is waiting for an I/O device to respond
- D. The process has not yet been created

Answer: A

Q270. What does the exit() system call do?

- A. Terminates the calling process and returns a status to the parent
- B. Restarts the operating system
- C. Closes all open files on the system
- D. Logs the user out of the system

Answer: A

Q271. What is the main goal of CPU scheduling?

- A. To maximize CPU utilization and ensure fair process execution
- B. To increase the physical clock speed of the CPU
- C. To add more cores to the processor
- D. To reduce the amount of installed RAM

Answer: A

Q272. What is a scheduling algorithm in operating systems?

- A. A method that determines which process gets the CPU next
- B. A formula for calculating memory addresses
- C. A technique for compressing files on disk
- D. A procedure for installing device drivers

Answer: A

Q273. In Round Robin scheduling, what happens when a process's time quantum expires?

- A. The process is preempted and moved to the end of the ready queue
- B. The process is terminated immediately
- C. The process continues running until it finishes
- D. The process is moved to a higher priority queue

Answer: A

Q274. What is a non-preemptive scheduling algorithm?

- A. An algorithm where a running process keeps the CPU until it voluntarily releases it
- B. An algorithm that can interrupt any process at any time
- C. An algorithm that runs all processes simultaneously
- D. An algorithm used only for real-time systems

Answer: A

Q275. What does response time measure in CPU scheduling?

- A. The time from submission of a request until the first response is produced
- B. The total time a process takes to complete execution
- C. The time a process spends waiting in the ready queue
- D. The time the CPU is idle between processes

Answer: A

Q276. Which scheduling algorithm is the simplest to implement?

- A. First-Come, First-Served (FCFS)
- B. Multilevel Feedback Queue
- C. Completely Fair Scheduler
- D. Rate Monotonic Scheduling

Answer: A

Q277. What is CPU burst in the context of process execution?

- A. A period during which a process is actively using the CPU for computation
- B. The maximum speed at which the CPU can operate
- C. A hardware failure that causes the CPU to overheat
- D. The time it takes to boot the CPU

Answer: A

Q278. In priority scheduling, what happens when two processes have the same priority?

- A. They are typically scheduled using FCFS among equal-priority processes
- B. The process with the larger PID always runs first
- C. Both processes run simultaneously on the same core
- D. One process is terminated to resolve the conflict

Answer: A

Q279. What is an I/O burst in process execution?

- A. A period during which a process waits for an I/O operation to complete
- B. A spike in CPU processing speed
- C. A burst of data transmitted over the network
- D. A sudden increase in available memory

Answer: A

Q280. What does the dispatcher module do in CPU scheduling?

- A. Gives control of the CPU to the process selected by the scheduler
- B. Selects which process should run next from the ready queue
- C. Calculates the priority of each process
- D. Manages memory allocation for processes

Answer: A

Q281. Which component is NOT shared between threads of the same process?

- A. Global variables
- B. Stack
- C. Code section
- D. Open file descriptors

Answer: B

Q282. What is the primary reason thread creation is faster than process creation?

- A. Threads use faster CPUs
- B. Threads share the address space and resources of their parent process
- C. Threads do not need an operating system
- D. Threads are stored on the hard disk

Answer: B

Q283. Which of the following best describes a single-threaded process?

- A. A process that can run on multiple CPUs simultaneously
- B. A process with exactly one thread of execution
- C. A process that cannot perform I/O operations
- D. A process that runs only in kernel mode

Answer: B

Q284. What is the role of a thread library?

- A. To manage hardware interrupts
- B. To provide an API for creating and managing threads
- C. To compile multithreaded programs
- D. To allocate disk space for threads

Answer: B

Q285. Context switching between threads of the same process is generally faster than between processes because:

- A. Threads have smaller program counters
- B. No need to switch the memory address space
- C. Threads do not use the CPU
- D. The operating system does not track threads

Answer: B

Q286. Which threading model maps many user-level threads to many kernel threads?

- A. One-to-one model
- B. Many-to-one model
- C. Many-to-many model
- D. One-to-many model

Answer: C

Q287. In Java, which interface must a class implement to define a task that can be executed by a thread?

- A. Serializable
- B. Runnable
- C. Callable only
- D. Throwable

Answer: B

Q288. What does the term 'multicore programming' refer to?

- A. Programming with multiple monitors
- B. Writing programs that can utilize multiple processing cores
- C. Running multiple operating systems simultaneously
- D. Using multiple keyboards for input

Answer: B

Q289. What is the difference between concurrency and parallelism in threading?

- A. They are identical concepts
- B. Concurrency means tasks make progress over time; parallelism means tasks run simultaneously
- C. Parallelism requires only one CPU core
- D. Concurrency is only possible with GPUs

Answer: B

Q290. Which register is unique to each thread and tracks the next instruction to execute?

- A. Base register
- B. Program counter
- C. Memory address register
- D. I/O register

Answer: B

Q291. What is the purpose of process synchronization?

- A. To make all processes run at the same speed
- B. To coordinate the execution of processes that share resources to prevent data inconsistency
- C. To synchronize the system clock with an external server
- D. To ensure all processes start at the same time

Answer: B

Q292. The 'progress' requirement for the critical section problem states that:

- A. All processes must execute at the same speed
- B. Only processes not in their remainder section can participate in deciding which process enters the critical section next
- C. A process must never leave the critical section
- D. All processes must enter the critical section simultaneously

- D. All processes must enter the critical section simultaneously

Answer: B

Q293. The 'bounded waiting' requirement for the critical section problem ensures that:

- A. The critical section has a fixed size
- B. There is a limit on the number of times other processes can enter the critical section after a process has made a request
- C. Processes wait for exactly 10 seconds
- D. Only bounded variables are used

Answer: B

Q294. What does a mutex lock provide?

- A. Unlimited concurrent access to shared data
- B. Mutual exclusion so that only one thread can hold the lock at a time
- C. Automatic parallelization of code
- D. Network synchronization between computers

Answer: B

Q295. What operation decreases the value of a semaphore?

- A. signal()
- B. post()
- C. wait()
- D. release()

Answer: C

Q296. What is a spinlock?

- A. A lock that puts the waiting thread to sleep
- B. A lock where the waiting thread continuously checks if the lock is available in a loop
- C. A lock that only works on single-core systems
- D. A lock that spins the hard disk for faster access

Answer: B

Q297. In the producer-consumer problem, what shared resource do the producer and consumer access?

- A. The CPU registers
- B. A bounded buffer
- C. The page table
- D. The file system journal

Answer: B

Q298. What is an atomic operation?

- A. An operation related to nuclear physics
- B. An operation that completes entirely without interruption
- C. An operation that can be partially executed
- D. An operation that requires multiple CPU cycles

Answer: B

Q299. What does the signal() operation do on a semaphore?

- A. Decreases its value by one
- B. Increases its value by one and may wake a waiting process
- C. Destroys the semaphore
- D. Resets the semaphore to zero

Answer: B

Q300. What is the key difference between a binary semaphore and a counting semaphore?

- A. Binary semaphores are faster
- B. A binary semaphore has only values 0 and 1, while a counting semaphore can range over an unrestricted domain
- C. Counting semaphores cannot be used for mutual exclusion
- D. Binary semaphores work only in user mode

Answer: B

Q301. Which of the following is a necessary condition for deadlock?

- A. Preemption of all resources
- B. Mutual exclusion
- C. Unlimited resources
- D. Single-threaded execution

Answer: B

Q302. In a resource allocation graph, what does an edge from a process to a resource indicate?

- A. The process has finished using the resource
- B. The process is requesting the resource
- C. The resource is allocated to the process
- D. The process has released the resource

Answer: B

Q303. What are the four necessary conditions for deadlock?

- A. Speed, memory, I/O, CPU
- B. Mutual exclusion, hold and wait, no preemption, circular wait
- C. Scheduling, paging, caching, buffering
- D. Authentication, authorization, encryption, logging

Answer: B

Q304. What is deadlock avoidance?

- A. Ignoring the possibility of deadlock
- B. Using additional information about future resource requests to ensure the system never enters an unsafe state
- C. Terminating all processes periodically
- D. Removing all shared resources from the system

Answer: B

Q305. What does deadlock recovery involve?

- A. Preventing deadlock from ever occurring
- B. Detecting that a deadlock has occurred and taking action to break it
- C. Ignoring the deadlock permanently
- D. Adding more resources to the system preemptively

Answer: B

Q306. An unsafe state means:

- A. Deadlock has already occurred
- B. The system may potentially lead to deadlock; there is no guarantee of a safe sequence
- C. All processes have terminated
- D. The system has run out of memory

Answer: B

Q307. In deadlock handling, what does the 'no preemption' condition mean?

- A. Resources can be forcibly taken from processes at any time
- B. Resources cannot be forcibly taken away from a process; they must be released voluntarily
- C. Processes cannot be scheduled on the CPU
- D. No process has higher priority than another

Answer: B

Q308. Which of the following is an example of a reusable resource?

- A. A CPU interrupt signal
- B. A printer or a memory block
- C. A network packet
- D. A one-time password

Answer: B

Q309. What is the simplest method to deal with deadlocks?

- A. Use the Banker's Algorithm
- B. Ignore the problem entirely (ostrich algorithm)
- C. Preempt all resources
- D. Allow only one process to run at a time

Answer: B

Q310. What type of graph is used specifically for deadlock detection with single-instance resources?

- A. A binary tree
- B. A wait-for graph
- C. A B+ tree
- D. A hash table

Answer: B

Q311. What is the primary function of a Memory Management Unit (MMU)?

- A. To increase the size of physical RAM
- B. To translate logical addresses to physical addresses at runtime
- C. To manage the file system
- D. To schedule CPU processes

Answer: B

Q312. What is a frame in the context of paging?

- A. A division of logical memory
- B. A fixed-size block of physical memory
- C. A type of CPU register
- D. A section of the hard disk

Answer: B

Q313. What is the difference between a logical address and a physical address?

- A. They are always the same value
- B. A logical address is generated by the CPU and a physical address is the actual location in memory hardware
- C. A physical address is virtual and a logical address is real
- D. Logical addresses are used only for disk storage

Answer: B

Q314. What is the purpose of a base register in memory management?

- A. To store the process's name
- B. To define the starting physical address of a process's memory space
- C. To count the number of pages
- D. To store the program counter

Answer: B

Q315. What is the role of the limit register in memory protection?

- A. It stores the maximum speed of memory access
- B. It specifies the size of the range of legal addresses for a process
- C. It limits the number of processes in the system
- D. It controls the clock speed of the CPU

Answer: B

Q316. What is dynamic loading?

- A. Loading all routines into memory at program start
- B. Loading a routine into memory only when it is called, reducing memory usage
- C. Loading programs onto the hard disk
- D. Loading the operating system into the GPU

Answer: B

Q317. What is the difference between first-fit and best-fit memory allocation?

- A. They always produce the same result
- B. First-fit allocates the first hole large enough; best-fit allocates the smallest hole large enough
- C. Best-fit is always slower than first-fit
- D. First-fit only works with paging

Answer: B

Q318. What problem does compaction solve?

- A. Internal fragmentation
- B. External fragmentation by shuffling memory contents to place all free memory together
- C. Page faults
- D. CPU cache misses

Answer: B

Q319. What does demand paging mean?

- A. All pages are loaded into memory at startup
- B. Pages are loaded into memory only when they are accessed and needed
- C. The CPU demands faster memory
- D. Pages are stored only on the hard disk

Answer: B

Q320. What is the valid-invalid bit in a page table entry used for?

- A. To indicate whether the data in the page has errors
- B. To indicate whether the page is currently in physical memory (valid) or not (invalid)
- C. To store the page number
- D. To count the number of accesses to the page

Answer: B

Q321. What is the purpose of a file system?

- A. To manage CPU scheduling
- B. To organize, store, and retrieve files on storage media
- C. To control network connections
- D. To manage power distribution to hardware

Answer: B

Q322. What is a relative path in a file system?

- A. A path that always starts from the root directory
- B. A path specified relative to the current working directory
- C. A path that points to a network resource
- D. A path that can only reference directories, not files

Answer: B

Q323. What file attribute indicates the number of bytes in a file?

- A. File type
- B. File size
- C. File name
- D. File permissions

Answer: B

Q324. What does the close() system call do in file operations?

- A. Deletes the file permanently
- B. Releases the file's resources and removes the entry from the open-file table
- C. Creates a backup of the file
- D. Changes the file's permissions

Answer: B

Q325. What is a directory entry?

- A. A CPU register that stores directory information
- B. A record in a directory that maps a file name to its location on storage
- C. A type of file that cannot be opened
- D. A network address for remote files

Answer: B

Q326. What is a single-level directory structure?

- A. A hierarchy of nested directories
- B. A single directory containing all files in the system
- C. A directory structure with exactly two levels
- D. A directory that stores only one file

Answer: B

Q327. What does 'mounting' a file system mean?

- A. Physically installing a hard drive
- B. Making a file system accessible by attaching it to a point in the existing directory tree
- C. Formatting a disk partition
- D. Encrypting the file system

Answer: B

Q328. What type of file access reads data sequentially from beginning to end?

- A. Random access
- B. Sequential access
- C. Indexed access
- D. Direct access

Answer: B

Q329. What is a two-level directory structure?

- A. A directory with two files
- B. Each user has a separate directory under a master file directory
- C. A directory stored on two different disks
- D. A directory accessible by only two users

Answer: B

Q330. What are file permissions used for?

- A. To compress files to save storage space
- B. To control which users can read, write, or execute a file
- C. To speed up file access times
- D. To encrypt file contents automatically

Answer: B

Q331. What is the main advantage of secondary storage over primary memory (RAM)?

- A. It is faster than RAM
- B. It is non-volatile and retains data when power is off
- C. It is directly accessible by the CPU
- D. It has lower latency than RAM

Answer: B

Q332. What is a cylinder in the context of hard disk drives?

- A. The motor that spins the disk
- B. The set of tracks at the same position across all platters
- C. A single sector on the disk
- D. The disk controller chip

Answer: B

Q333. What is the FCFS (First-Come, First-Served) disk scheduling algorithm?

- A. Requests are served based on their distance from the current head position
- B. Requests are served in the order they arrive in the queue
- C. Requests are served from the innermost track to the outermost track
- D. Requests are served based on their priority

Answer: B

Q334. Why do SSDs have no seek time or rotational latency?

- A. They use faster motors
- B. They have no moving parts; data is accessed electronically from flash memory cells
- C. They use optical technology instead of magnetic
- D. They spin at infinite speed

Answer: B

Q335. What does RAID stand for?

- A. Random Array of Independent Data
- B. Redundant Array of Independent Disks
- C. Rapid Access to Internal Drives
- D. Reliable Access to Indexed Data

Answer: B

Q336. What is the purpose of disk formatting?

- A. To make the disk spin faster
- B. To prepare a disk for use by creating the necessary data structures for a file system
- C. To encrypt all data on the disk
- D. To physically clean the disk platters

Answer: B

Q337. What is the function of the disk controller?

- A. To physically manufacture the disk
- B. To interface between the disk drive and the computer, managing data transfer and disk operations
- C. To supply power to the disk
- D. To cool the disk platters

Answer: B

Q338. What is swap space used for?

- A. Swapping data between two hard drives
- B. Extending virtual memory by storing pages that are not currently in physical RAM
- C. Exchanging files between users
- D. Temporarily storing CPU registers

Answer: B

Q339. What is a platter in a hard disk drive?

- A. The cable connecting the disk to the motherboard
- B. A circular, flat disk coated with magnetic material that stores data
- C. The electronic circuit board of the drive
- D. The protective case of the drive

Answer: B

Q340. What is an I/O port?

- A. A physical port for connecting displays
- B. A set of registers through which the CPU communicates with an I/O device controller
- C. A network port used for internet connections
- D. A type of RAM used for I/O caching

Answer: B

Q341. What is the main disadvantage of polling (programmed I/O)?

- A. It does not work with any devices
- B. The CPU wastes cycles repeatedly checking the device status instead of doing useful work
- C. It is too fast for most devices
- D. It requires special hardware

Answer: B

Q342. How does DMA (Direct Memory Access) benefit the system?

- A. It eliminates the need for memory entirely
- B. It allows devices to transfer data directly to or from memory without CPU involvement for each byte
- C. It doubles the CPU's clock speed during I/O
- D. It replaces the operating system's I/O management

Answer: B

Q343. What is spooling in the context of I/O?

- A. Rewinding magnetic tape
- B. Storing output data in a buffer (usually on disk) for a device that cannot accept interleaved data from multiple sources
- C. Speeding up I/O operations
- D. Connecting multiple printers together

Answer: B

Q344. What is the difference between a block device and a character device?

- A. Block devices are slower than character devices
- B. Block devices transfer data in fixed-size blocks and support random access; character devices transfer data one character at a time in a stream
- C. Character devices can only handle text files
- D. Block devices do not use device drivers

Answer: B

Q345. What is an interrupt in the context of I/O?

- A. A power outage affecting the system
- B. A signal from a device to the CPU indicating that an I/O event needs attention
- C. A type of memory error
- D. A command from the CPU to stop a device

Answer: B

Q346. What role does a device driver play in the I/O system?

- A. It physically connects devices to the computer
- B. It provides a uniform interface for the OS to interact with a specific hardware device
- C. It generates electricity for the device
- D. It manufactures new devices

Answer: B

Q347. What is I/O buffering?

- A. Removing data from I/O devices
- B. Temporarily storing data in memory during transfer between a device and an application to handle speed differences
- C. Encrypting I/O data for security
- D. Compressing data before sending it to a device

Answer: B

Q348. What is a bus in a computer system?

- A. A vehicle for transporting computer equipment
- B. A shared communication pathway that transfers data between components of a computer
- C. A type of secondary storage
- D. A software program for managing I/O

Answer: B

Q349. What happens when an interrupt occurs during program execution?

- A. The program is permanently deleted
- B. The CPU saves the current state and transfers control to the appropriate interrupt handler routine
- C. The computer shuts down
- D. The interrupt is ignored until the program finishes

Answer: B

Q350. What is authorization in operating system security?

- A. Verifying a user's identity
- B. Determining what resources a verified user is permitted to access
- C. Encrypting user data
- D. Logging user activity

Answer: B

Q351. What is a worm in computer security?

- A. A program that modifies other programs by inserting its code
- B. A self-replicating program that spreads across networks without needing a host program
- C. A program that disguises itself as legitimate software
- D. A hardware device that protects against malware

Answer: B

Q352. What is a denial-of-service (DoS) attack?

- A. Stealing data from a computer
- B. An attack that overwhelms a system's resources to make it unavailable to legitimate users
- C. Installing malware on a computer
- D. Breaking encryption to access protected data

Answer: B

Q353. What is the purpose of a user account password?

- A. To encrypt files automatically
- B. To authenticate the user's identity before granting system access
- C. To speed up the login process
- D. To store the user's preferences

Answer: B

Q354. What is the difference between a virus and a Trojan horse?

- A. Viruses are hardware; Trojans are software
- B. A virus self-replicates by attaching to other programs; a Trojan disguises itself as legitimate software but does not self-replicate
- C. A Trojan self-replicates; a virus does not
- D. There is no difference between them

Answer: B

Q355. What does the principle of least privilege mean in practice?

- A. Users should have the maximum possible privileges
- B. Users and programs should only have the minimum privileges necessary to complete their tasks
- C. All users should have equal privileges
- D. Privileges should be assigned randomly

Answer: B

Q356. What is the purpose of a security audit log?

- A. To speed up system performance
- B. To record security-relevant events for later review and analysis
- C. To encrypt system files
- D. To authenticate users automatically

Answer: B

Q357. What is biometric authentication?

- A. Authentication using passwords only
- B. Authentication using unique physical characteristics such as fingerprints, facial recognition, or iris patterns
- C. Authentication using security tokens only
- D. Authentication using encryption keys

Answer: B

Q358. What is a protection domain in operating system security?

- A. A geographic region where the computer is located
- B. A set of access rights that defines what resources a process can access and what operations it can perform
- C. A physical barrier around the server room
- D. A type of antivirus software

Answer: B

Q359. What is the function of a firewall?

- A. To cool down the CPU to prevent overheating
- B. To filter network traffic based on security rules, blocking unauthorized access while permitting legitimate communication
- C. To encrypt all files on the hard drive
- D. To scan files for viruses

Answer: B

Q360. What is the relationship between disk access time and its components?

- A. Access time equals seek time only
- B. Access time is the sum of seek time, rotational latency, and data transfer time
- C. Access time equals rotational latency only
- D. Access time equals transfer time minus seek time

Answer: B

Medium Questions

360 questions

Q361. What is the difference between multiprogramming and multitasking?

- A. Multiprogramming uses multiple CPUs; multitasking one
- B. Multiprogramming maximizes CPU; multitasking adds sharing
- C. There is no meaningful difference between them
- D. Multitasking is an older technique than multiprogramming

Answer: B

Q362. What is a system call?

- A. A call made from one user to another user
- B. A function call within user-level programs
- C. A programmatic interface to request OS services
- D. A type of hardware interrupt signal to CPU

Answer: C

Q363. Which mode of CPU operation has unrestricted access to all hardware resources?

- A. User mode
- B. Safe mode
- C. Debug mode
- D. Kernel mode

Answer: D

Q364. What is the purpose of the mode bit in the CPU?

- A. To store the current instruction address
- B. To manage the CPU cache memory size
- C. To count the total CPU clock cycles
- D. To distinguish user mode from kernel mode

Answer: D

Q365. Spooling in operating systems refers to:

- A. A type of interrupt handling method
- B. Overlapping I/O with CPU computation
- C. A type of CPU scheduling algorithm
- D. A memory management technique used

Answer: B

Q366. An interrupt is:

- A. A type of software error in code
- B. A signal needing immediate CPU attention
- C. A specific type of system call request
- D. A process scheduling technique in OS

Answer: B

Q367. What is the difference between a trap and an interrupt?

- A. A trap is software-generated, interrupt is hardware
- B. There is no real difference between them
- C. A trap executes faster than an interrupt does
- D. A trap is hardware-generated, interrupt is software

Answer: A

Q368. DMA (Direct Memory Access) is used to:

- A. Speed up internal CPU processing tasks
- B. Transfer data to memory without the CPU
- C. Increase the processor clock speed rate
- D. Manage virtual memory paging on disk

Answer: B

Q369. A distributed operating system:

- A. Makes independent computers appear as one system
- B. Runs only on a single standalone machine
- C. Cannot share resources between multiple machines
- D. Is the same as a standard network operating system

Answer: A

Q370. What is the bootstrap program?

- A. A program that manages user accounts and groups
- B. A program that handles active network connections
- C. A program that compiles other user-level programs
- D. The first program run at power-on to load the OS

Answer: D

Q371. In a microkernel OS, which of the following typically runs in user space?

- A. Interrupt handling routines
- B. Core CPU scheduling
- C. Hardware-level memory mgmt
- D. File system services

Answer: D

Q372. What is the primary mechanism microkernels use for communication between services?

- A. Message passing IPC
- B. Shared memory only
- C. Continuous polling
- D. Direct function calls

Answer: A

Q373. System calls are typically invoked by user programs through:

- A. Direct hardware manipulation
- B. Application Programming APIs
- C. Bootloader entry points
- D. Device driver interfaces

Answer: B

Q374. What is the main disadvantage of the layered approach to OS design?

- A. It does not provide any file system support
- B. Difficulty defining layers and traversal overhead
- C. It is too simple and basic to be practical
- D. It cannot support any form of multitasking

Answer: B

Q375. What is a loadable kernel module (LKM)?

- A. Code loaded into the kernel at runtime dynamically
- B. A user application loaded on demand by the OS
- C. A type of system call for interprocess messaging
- D. A bootloader component for starting the OS kernel

Answer: A

Q376. Which of the following is a category of system calls?

- A. Compilation control
- B. Code optimization
- C. Information maintenance
- D. Syntax analysis

Answer: C

Q377. Parameters to system calls can be passed by placing them in:

- A. Only through CPU registers
- B. Only via the system stack
- C. Registers, stack, or memory block
- D. Only through shared memory

Answer: C

Q378. The POSIX API is:

- A. A Windows-specific system API
- B. A file format specification
- C. A portable UNIX interface standard
- D. A networking protocol standard

Answer: C

Q379. What is the purpose of the system call interface?

- A. To link API calls to kernel implementations
- B. To format disks and storage partitions
- C. To manage user passwords and accounts
- D. To execute hardware instructions directly

Answer: A

Q380. Windows NT uses which type of kernel architecture?

- A. Hybrid kernel
- B. Pure microkernel
- C. Pure monolithic
- D. Exokernel type

Answer: A

Q381. In the producer-consumer problem using shared memory, what role does the buffer play?

- A. It handles hardware interrupt processing
- B. It is a shared region for producer-consumer
- C. It stores the executable program code
- D. It manages the CPU scheduling algorithm

Answer: B

Q382. What is the difference between shared memory and message passing IPC?

- A. Shared memory faster for large data; message passing for remote
- B. There is no difference between these two approaches
- C. Shared memory uses common region; message passing uses kernel
- D. Both B and C are correct descriptions of their differences

Answer: D

Q383. What happens during a context switch?

- A. Current process state is saved, next is loaded
- B. All of main memory is cleared completely
- C. All running processes are terminated at once
- D. The CPU is shut down and then restarted

Answer: A

Q384. In UNIX, what does the exec() family of system calls do?

- A. Terminates the currently running process
- B. Replaces process image with new program
- C. Suspends the process until a signal
- D. Creates a brand new child process

Answer: B

Q385. What is cascading termination?

- A. Terminating processes one by one sequentially
- B. OS terminates all children when parent exits
- C. Restarting processes in a specific sequence
- D. Suspending processes in a particular order

Answer: B

Q386. A pipe in UNIX provides:

- A. Bidirectional communication between any processes
- B. Network communication across remote machines
- C. Direct communication with the hardware devices
- D. Unidirectional communication between parent-child

Answer: D

Q387. What is a named pipe (FIFO) and how does it differ from an ordinary pipe?

- A. It only works on Microsoft Windows platforms
- B. It provides faster data communication speed
- C. It has a name and works between unrelated processes
- D. It is identical to an ordinary pipe

Answer: C

Q388. Which scheduling queue holds processes that are in main memory and ready to execute?

- A. Wait queue
- B. Device queue
- C. Ready queue
- D. Job queue

Answer: C

Q389. The long-term scheduler (job scheduler) controls:

- A. Which process gets I/O device access next
- B. Which process gets swapped out to the disk
- C. Which process gets the CPU execution next
- D. Which processes are admitted to the system

Answer: D

Q390. What is the role of the short-term scheduler (CPU scheduler)?

- A. To swap processes between memory and disk
- B. To select which ready process gets the CPU
- C. To admit new jobs into the system queue
- D. To terminate processes that have finished

Answer: B

Q391. What is the convoy effect in FCFS scheduling?

- A. Processes are grouped into convoys for batch runs
- B. Short processes wait for long ones, raising wait time
- C. CPU utilization increases dramatically and suddenly
- D. All processes finish at exactly the same time

Answer: B

Q392. In Shortest Remaining Time First (SRTF) scheduling, what happens when a new process arrives?

- A. The current process is always preempted regardless of timing
- B. The new process always waits until the current one completes
- C. The scheduler compares remaining times and preempts if shorter
- D. The new process automatically gets the highest priority level

Answer: C

Q393. What is starvation in the context of CPU scheduling?

- A. A process running completely out of allocated memory
- B. A process consuming too much CPU time from the system
- C. A deadlock situation between two or more blocked tasks
- D. A low-priority process waiting indefinitely for the CPU

Answer: D

Q394. Aging is a technique used to:

- A. Track how long each process has been running on the CPU
- B. Increase process priority over time to prevent starvation
- C. Decrease the priority of a process as it runs much longer
- D. Remove old processes from the system to free up resources

Answer: B

Q395. If the time quantum in Round Robin is very large, it behaves like:

- A. Multilevel queues
- B. Priority scheduling
- C. SJF scheduling
- D. FCFS scheduling

Answer: D

Q396. If the time quantum in Round Robin is very small, what problem occurs?

- A. Excessive switching
- B. System deadlocking
- C. Memory overflowing
- D. Process starvation

Answer: A

Q397. In Multilevel Queue Scheduling, processes are:

- A. Permanently assigned to queues by type
- B. All placed in one single queue
- C. Moved between queues based on priority
- D. Randomly assigned to available queues

Answer: A

Q398. What distinguishes Multilevel Feedback Queue from Multilevel Queue scheduling?

- A. It only uses the Round Robin algorithm
- B. Processes can move between queues by behavior
- C. It does not support priority scheduling
- D. It uses fewer scheduling queues overall

Answer: B

Q399. Dispatch latency is:

- A. The time a process waits in the ready queue
- B. The time between two consecutive CPU interrupts
- C. The time to stop one process and start another
- D. The total execution time of a single process

Answer: C

Q400. Response time in scheduling is defined as:

- A. The time between two context switches
- B. The time to complete all system processes
- C. The total CPU execution time of process
- D. Time from submission until first response

Answer: D

Q401. What is the many-to-one threading model?

- A. Each user thread maps to one kernel thread
- B. Many kernel threads map to one user thread
- C. Many user-level threads map to one kernel thread
- D. One user thread maps to many kernel threads

Answer: C

Q402. What is the one-to-one threading model?

- A. One process only ever has one thread
- B. One kernel thread maps to many user threads
- C. One user thread maps to one kernel thread
- D. All threads share one single CPU together

Answer: C

Q403. What is the disadvantage of the many-to-one threading model?

- A. If one thread blocks, all threads in process block
- B. It cannot create more than two threads total
- C. It uses too many kernel resources overall
- D. It is too slow for practical usage scenarios

Answer: A

Q404. What is a thread pool?

- A. A memory allocation area used for threads
- B. Pre-created threads that wait for work tasks
- C. A collection of currently unused threads
- D. A debugging tool for analyzing thread bugs

Answer: B

Q405. What happens to threads when fork() is called in a multithreaded program?

- A. The child process is created with no threads
- B. All threads are always duplicated in the child
- C. Fork cannot be used in multithreaded programs
- D. It depends on the system; some dup all, some one

Answer: D

Q406. What is thread cancellation?

- A. Terminating a thread before completion
- B. Creating a brand new thread from scratch
- C. Moving a thread to a different process
- D. Suspending a thread temporarily in place

Answer: A

Q407. What is the difference between asynchronous and deferred thread cancellation?

- A. Asynchronous is safer because it checks all resources before acting
- B. Asynchronous terminates immediately; deferred checks at safe points
- C. Deferred cancellation is always much faster than asynchronous type
- D. There is no difference between the two cancellation types

Answer: B

Q408. Thread-Local Storage (TLS) allows:

- A. Threads to communicate with each other via messages
- B. Each thread to have its own copy of certain data
- C. All threads to share the exact same data copies
- D. Data to be stored persistently on disk for threads

Answer: B

Q409. What is implicit threading?

- A. Threads that remain invisible to the programmer
- B. Threads that run implicitly without any OS scheduling
- C. Threads that do not use any system resources at all
- D. Transferring thread management to compilers and runtimes

Answer: D

Q410. OpenMP is used for:

- A. File compression utilities
- B. Parallel shared-memory programming
- C. Database management systems
- D. Network programming via APIs

Answer: B

Q411. What is Peterson's solution for the critical section problem?

- A. A solution implemented using monitors
- B. A hardware-based solution only
- C. A solution implemented using semaphores
- D. A software solution using turn and flag

Answer: D

Q412. The test-and-set instruction is:

- A. A specific type of semaphore implementation
- B. An atomic hardware instruction to test and set
- C. A scheduling algorithm for critical sections
- D. A software-only algorithm for locking

Answer: B

Q413. The compare-and-swap (CAS) instruction:

- A. Compares two files and swaps their contents
- B. Compares the speeds of two different CPUs
- C. Atomically compares and swaps a memory value
- D. Swaps two processes in the scheduling queue

Answer: C

Q414. A monitor is:

- A. A specific type of counting semaphore lock mechanism
- B. A construct encapsulating shared data with mutual exclusion
- C. A hardware lock on the system memory bus interface
- D. A display device or screen monitor unit

Answer: B

Q415. Condition variables in monitors are used with which operations?

- A. push() and pop()
- B. acquire() and release()
- C. wait() and signal()
- D. lock() and unlock()

Answer: C

Q416. In the bounded buffer problem, what are the semaphores typically used?

- A. One counting semaphore for the buffer
- B. Only a single mutex lock for the buffer
- C. Mutex plus empty and full semaphores
- D. Two binary semaphores for the buffer

Answer: C

Q417. In the readers-writers problem, what is the key synchronization challenge?

- A. Only one reader can read the data at a time
- B. Only one process can ever access the shared database
- C. Writers can write while readers are actively reading
- D. Multiple readers can read; writers need exclusive access

Answer: D

Q418. What is a spinlock most useful for?

- A. Short critical sections on multiprocessor systems
- B. Any critical section on single processor only
- C. I/O-bound processes waiting for disk access
- D. Long critical sections on single processors

Answer: A

Q419. What is priority inversion?

- A. Changing process priorities dynamically at runtime
- B. Assigning priorities in reverse numerical order
- C. A scheduling bug in all operating system kernels
- D. High-priority blocked by low-priority holding resource

Answer: D

Q420. Priority inheritance protocol solves priority inversion by:

- A. Raising low-priority to match highest waiting
- B. Blocking all medium-priority from running
- C. Terminating the low-priority resource holder
- D. Removing all priorities from every process

Answer: A

Q421. A safe state is one where:

- A. A safe sequence exists for all processes
- B. No process is waiting for any resource
- C. All system resources are fully available
- D. No process is currently running at all

Answer: A

Q422. What is the relationship between safe state and deadlock?

- A. An unsafe state always guarantees that deadlock exists
- B. Safe state means deadlock currently exists in system
- C. Safe and unsafe states are unrelated to deadlock at all
- D. Safe means no deadlock; unsafe means deadlock may occur

Answer: D

Q423. To prevent the 'hold and wait' condition, a process must:

- A. Use only shared resources that do not need locking
- B. Never request any resources from the operating system
- C. Request all resources at once, or release before requesting
- D. Hold all needed resources indefinitely without releasing

Answer: C

Q424. To prevent circular wait, one approach is:

- A. Remove all shared resources from system
- B. Allow processes to request in any order
- C. Impose ordering and require increasing order
- D. Use only a single resource type allowed

Answer: C

Q425. In the Banker's Algorithm, the 'Need' matrix represents:

- A. Currently allocated resources to each process
- B. The total number of resources in whole system
- C. Remaining resources each process may still need
- D. Total resources currently available to allocate

Answer: C

Q426. What is deadlock detection?

- A. Allowing deadlocks then detecting and recovering
- B. Ignoring deadlocks completely in all situations
- C. Avoiding deadlocks using the Banker Algorithm
- D. Preventing deadlocks from occurring in system

Answer: A

Q427. For deadlock detection with single-instance resources, which algorithm is used?

- A. Wait-for graph cycling
- B. Banker's Algorithm test
- C. Round Robin scheduling
- D. Best Fit allocating

Answer: A

Q428. Which deadlock recovery method terminates processes?

- A. Resource preemption from deadlocked processes
- B. Restarting the entire system from the beginning
- C. Ignoring the deadlock and hoping it resolves
- D. Process termination - abort all or one at a time

Answer: D

Q429. When recovering from deadlock by process termination, which factor is NOT typically considered?

- A. Time it has been running
- B. Process priority level
- C. Resources it now holds
- D. The color of terminal

Answer: D

Q430. The ostrich algorithm for handling deadlocks refers to:

- A. Ignoring deadlocks when rare and costly to handle
- B. A recovery mechanism for resolving deadlocks
- C. A sophisticated deadlock detection algorithm
- D. A prevention technique eliminating conditions

Answer: A

Q431. The Translation Lookaside Buffer (TLB) is:

- A. A disk cache for swap file contents
- B. A fast cache for page table entries
- C. A type of random access memory chip
- D. A backup copy of the main page table

Answer: B

Q432. In demand paging, pages are loaded into memory:

- A. All at once when the process first starts
- B. Based on a predetermined schedule
- C. Only when they are actually referenced
- D. Randomly without any specific order

Answer: C

Q433. The FIFO page replacement algorithm replaces:

- A. The page that has been in memory longest
- B. The least recently used page in memory
- C. The most frequently used page in memory
- D. A random page selected from the frames

Answer: A

Q434. The LRU (Least Recently Used) page replacement algorithm replaces:

- A. The page that was loaded into memory first
- B. The page not used for the longest time period
- C. The largest page currently loaded in a frame
- D. The page with the lowest assigned page number

Answer: B

Q435. Belady's anomaly states that:

- A. More frames can increase faults for some algorithms
- B. More memory always reduces total page faults
- C. LRU always outperforms FIFO in every situation
- D. Optimal page replacement causes most page faults

Answer: A

Q436. The optimal page replacement algorithm (OPT) replaces:

- A. The page unused for longest future time
- B. The smallest page currently in a frame
- C. The least recently used page in memory
- D. The first page that was loaded in memory

Answer: A

Q437. Thrashing occurs when:

- A. The CPU is idle and not running tasks
- B. Memory is completely free and available
- C. A process pages more than it executes
- D. Too few processes are currently running

Answer: C

Q438. Segmentation divides a program into:

- A. Equal-sized partitions of memory
- B. Fixed-size pages of equal length
- C. Variable-size segments like code and data
- D. Random blocks of varying sizes

Answer: C

Q439. The first-fit memory allocation algorithm:

- A. Allocates memory blocks at random
- B. Searches for the smallest suitable hole
- C. Allocates the first hole large enough
- D. Allocates the largest available hole

Answer: C

Q440. The best-fit memory allocation algorithm:

- A. Is the fastest allocation algorithm
- B. Allocates the first hole it finds
- C. Allocates the largest available hole
- D. Searches for the smallest sufficient hole

Answer: D

Q441. What is the difference between contiguous, linked, and indexed file allocation?

- A. They only differ in the speed of read and write operations
- B. Contiguous uses adjacent; linked chains; indexed uses pointers
- C. They only differ in the security settings for file access
- D. They are all identical in behavior and performance

Answer: B

Q442. Free space on disk can be managed using:

- A. Only the FAT table for tracking
- B. Only bit vectors for tracking
- C. Only linked lists for tracking
- D. Bit vectors, lists, or counting

Answer: D

Q443. A tree-structured directory organizes files in:

- A. A randomly organized structure
- B. A flat single-level structure
- C. A hierarchical parent-child structure
- D. A circular connected structure

Answer: C

Q444. An acyclic-graph directory structure allows:

- A. Only one single link per file in the system
- B. Circular file references between directories
- C. No file sharing between any user accounts
- D. Sharing files via links while preventing cycles

Answer: D

Q445. What is the Virtual File System (VFS) layer?

- A. An abstraction for uniform file system interface
- B. A file system stored in virtual memory
- C. A compressed file system saving disk space
- D. A file system designed for virtual machines

Answer: A

Q446. NTFS (New Technology File System) is primarily used by:

- A. Linux
- B. macOS
- C. Android
- D. Windows

Answer: D

Q447. ext4 is:

- A. A database system format
- B. The default Linux file system
- C. A Windows file system format
- D. A macOS file system format

Answer: B

Q448. File system consistency checking is needed because:

- A. Users make frequent file handling errors
- B. Files grow too large for disk capacity
- C. Hardware never fails in normal practice
- D. A crash during ops leaves inconsistent state

Answer: D

Q449. Access control lists (ACLs) in file systems provide:

- A. File versioning for tracking changes over time
- B. File data encryption for security and privacy
- C. File data compression for saving disk space
- D. Fine-grained permissions for users and groups

Answer: D

Q450. Direct access (random access) to a file means:

- A. Reading or writing any block directly without prior reads
- B. Reading the file sequentially from start to end
- C. Accessing the file at random without any specific pattern
- D. Accessing the file exclusively through the OS kernel layer

Answer: A

Q451. The SCAN (elevator) disk scheduling algorithm:

- A. Moves head one direction servicing, then reverses
- B. Only scans tracks that have pending I/O requests
- C. Scans files for viruses and malware threats
- D. Scans all tracks in a random selection order

Answer: A

Q452. The C-SCAN (Circular SCAN) algorithm differs from SCAN in that:

- A. It only works on solid state drives not on HDDs
- B. It is slower than the standard SCAN algorithm is
- C. Returns to start without servicing for uniform wait
- D. It never reverses the head direction at all

Answer: C

Q453. The LOOK disk scheduling algorithm differs from SCAN in that:

- A. It is significantly slower than the SCAN algorithm
- B. Reverses when no more requests ahead, not at end
- C. It only handles one single request at a time
- D. It looks at all pending requests before starting

Answer: B

Q454. Why is disk scheduling less important for SSDs compared to HDDs?

- A. SSDs use a completely different data interface
- B. SSDs are actually slower than HDDs overall
- C. SSDs have no moving parts, so seek is negligible
- D. SSDs cannot handle multiple concurrent requests

Answer: C

Q455. RAID 0 provides:

- A. Data redundancy only for safety
- B. Performance via striping, no redundancy
- C. Mirroring of data across two disks
- D. Parity-based protection from failure

Answer: B

Q456. RAID 1 provides:

- A. No data protection of any kind
- B. Striping data across disks only
- C. Parity-based protection on disks
- D. Mirroring via duplicate on disks

Answer: D

Q457. RAID 5 uses:

- A. Double parity across all the disks
- B. Block striping with distributed parity
- C. No redundancy of any kind at all
- D. Only mirroring for redundancy

Answer: B

Q458. Low-level (physical) formatting creates:

- A. Logical partitions and volume table structures
- B. A high-level file system and all its metadata
- C. Sector structure with headers and error correction
- D. Directories and user files on the disk surface

Answer: C

Q459. A swap space on a disk is used for:

- A. Disk defragmentation temporary workspace
- B. Exchanging files between different users
- C. Temporary file storage for applications
- D. Storing pages swapped out from memory

Answer: D

Q460. The boot block on a disk contains:

- A. Bootstrap code to load OS
- B. User data and documents
- C. Error correction for sectors
- D. File system metadata tables

Answer: A

Q461. What is the difference between programmed I/O, interrupt-driven I/O, and DMA?

- A. Programmed polls; interrupt frees CPU; DMA bypasses CPU
- B. They only work with specific hardware device types
- C. They only differ in speed of the data transfer
- D. They are all identical in behavior and speed

Answer: A

Q462. What is memory-mapped I/O?

- A. Storing I/O data into a temporary buffer
- B. Mapping device registers to memory addresses
- C. A type of virtual memory for page tables
- D. Mapping files from disk into virtual memory

Answer: B

Q463. What is the purpose of the I/O subsystem in the OS kernel?

- A. To manage user accounts and authentication
- B. To compile programs into executable binaries
- C. To provide games and multimedia for users
- D. Uniform interfaces for buffering and scheduling

Answer: D

Q464. Double buffering uses:

- A. No buffers needed at all ever
- B. Three buffers in a rotation cycle
- C. One buffer for all data transfers
- D. Two buffers: fill one, process other

Answer: D

Q465. Caching in the I/O system differs from buffering in that:

- A. They are identical in purpose and behavior
- B. Buffering stores during transfer; cache keeps copies
- C. Caching is always slower than buffering in practice
- D. Buffering is used only for output data operations

Answer: B

Q466. What is a device controller's role?

- A. To manage hardware and translate OS commands
- B. To manage virtual memory paging and swapping
- C. To compile device driver source code files
- D. To schedule processes for CPU execution time

Answer: A

Q467. What is the difference between blocking and non-blocking I/O?

- A. Blocking is always faster than non-blocking I/O
- B. They are identical in every way and behavior
- C. Non-blocking prevents all I/O from starting
- D. Blocking suspends until done; non-blocking returns

Answer: D

Q468. What is the purpose of an interrupt controller?

- A. To block all incoming interrupt signals
- B. To generate interrupt signals to CPU
- C. To manage, prioritize, route interrupts
- D. To count total number of interrupts

Answer: C

Q469. What is the unified buffer cache?

- A. A cache located inside the CPU for all instructions
- B. Single cache for files and mapped memory, no duplication
- C. A cache shared by all system users equally
- D. A universal cache for every device in the system

Answer: B

Q470. Error handling in I/O systems typically involves:

- A. Shutting down the entire system on any error
- B. Deleting the faulty device from the system table
- C. Returning codes, retrying, logging, informing users
- D. Ignoring all errors that occur during I/O

Answer: C

Q471. An access control matrix is:

- A. A type of encryption algorithm for data security
- B. A type of visual display device for output
- C. A hardware protection mechanism in the CPU unit
- D. A model specifying subject access rights to objects

Answer: D

Q472. Access Control Lists (ACLs) are a column-based implementation of the access control matrix where:

- A. Each object stores subjects and their access rights
- B. A central authority stores all access control rights
- C. Access rights are not stored anywhere in the system
- D. Each subject stores its own access rights list

Answer: A

Q473. Capability lists are a row-based implementation of the access control matrix where:

- A. Capabilities are stored in a central directory
- B. Each object stores its own capability tokens
- C. Each subject holds objects and allowed operations
- D. Capabilities are not used in any modern system

Answer: C

Q474. A denial-of-service (DoS) attack:

- A. Modifies system files to gain higher access levels
- B. Steals confidential data from the system quietly
- C. Overwhelms a system to deny service to users
- D. Encrypts user files and demands ransom payment

Answer: C

Q475. A buffer overflow attack exploits:

- A. Too much available memory in the system overall
- B. Overwriting memory beyond a buffer boundary
- C. Network buffer limits causing packet loss
- D. Disk buffer overflow causing data corruption

Answer: B

Q476. In UNIX file permissions, the permission mode 755 means:

- A. Owner has rwx, group and others have r-x access
- B. Everyone has full unrestricted access to file
- C. The file is hidden from all users in system
- D. Only the owner can access and modify the file

Answer: A

Q477. What is the difference between symmetric and asymmetric encryption?

- A. Asymmetric is faster than symmetric in practice
- B. Symmetric uses one key; asymmetric uses key pair
- C. They are the same algorithm for encryption use
- D. Symmetric is always more secure than asymmetric

Answer: B

Q478. What is a protection domain?

- A. A set of access rights for process resources
- B. A physical area in the server room building
- C. A network domain managed by a DNS server
- D. A type of firewall zone for network security

Answer: A

Q479. Multifactor authentication combines:

- A. Using multiple different usernames
- B. Combining different auth factor types
- C. Allowing multiple login attempts
- D. Using multiple passwords for login

Answer: B

Q480. What is the role of a digital certificate in security?

- A. To encrypt files for security
- B. To manage user passwords securely
- C. To bind public key to identity via CA
- D. To compress data for efficient storage

Answer: C

Q481. What distinguishes a microkernel architecture from a monolithic kernel?

- A. Microkernels run all services in kernel space for speed
- B. Microkernels move most services to user space processes
- C. Monolithic kernels keep only minimal services in kernel
- D. Both architectures are identical in design and approach

Answer: B

Q482. What is the purpose of dual-mode operation in an operating system?

- A. To double the processing speed of the CPU cores
- B. To protect the system from errant user programs
- C. To run two operating systems simultaneously here
- D. To allow two separate users to log in together

Answer: B

Q483. How does a hypervisor differ from a traditional operating system?

- A. A hypervisor is just another name for an OS kernel
- B. A hypervisor only runs on mobile phone platforms
- C. A hypervisor manages virtual machines on hardware
- D. A hypervisor cannot manage hardware resources ever

Answer: C

Q484. What is the key advantage of time-sharing over batch processing?

- A. Batch processing provides faster interactive response
- B. Time-sharing eliminates the need for any memory
- C. Time-sharing provides interactive response to users
- D. Time-sharing consumes less CPU than batch does here

Answer: C

Q485. What role does the DMA controller play in an operating system?

- A. It compiles source code into machine language
- B. It schedules processes for execution on processors
- C. It manages user authentication and access control
- D. It transfers data between memory and I/O directly

Answer: D

Q486. Which statement about system programs is correct?

- A. They replace the need for an operating system here
- B. They provide convenient environment for development
- C. System programs only include text editor programs
- D. System programs are the same as the OS kernel

Answer: B

Q487. What is the trap instruction used for in operating systems?

- A. To physically trap faulty hardware in the system
- B. To prevent users from logging into the system
- C. To switch from user mode to kernel mode safely
- D. To catch and eliminate computer virus threats

Answer: C

Q488. What is the difference between policy and mechanism in OS design?

- A. They are interchangeable terms with the same meaning
- B. Policy refers to hardware; mechanism to software here
- C. Mechanism decides what; policy decides how it is done
- D. Policy decides what to do; mechanism decides how

Answer: D

Q489. What happens during a context switch in an operating system?

- A. The computer restarts and reloads the entire system
- B. All running processes are terminated and then removed
- C. The OS saves one process state and loads another
- D. The kernel is replaced with a newer version of code

Answer: C

Q490. What is the purpose of the OS system call interface?

- A. It controls voltage supply to all peripheral devices
- B. It connects the monitor cable directly to CPU chip
- C. It manages the physical layout of the motherboard
- D. It provides a layer between user programs and kernel

Answer: D

Q491. What is the main advantage of a layered OS structure?

- A. It eliminates the need for any system call interfaces
- B. It provides the fastest possible system performance now
- C. It requires less memory than all other architectures
- D. Each layer can be debugged and verified independently

Answer: D

Q492. How do loadable kernel modules benefit the operating system?

- A. They allow extending kernel functionality at runtime easily
- B. They eliminate all system calls from the OS call interface
- C. They require the entire kernel to be recompiled each time
- D. They replace the need for any device drivers completely now

Answer: A

Q493. What is the role of the system call interface in OS architecture?

- A. It only manages file system operations and nothing else
- B. It replaces the kernel in all microkernel architectures
- C. It serves as boundary between user programs and the OS
- D. It provides direct hardware access to user-level programs

Answer: C

Q494. What is a key disadvantage of the simple monolithic structure?

- A. It cannot access any hardware devices at all currently
- B. It is very slow due to many excessive mode switches
- C. It does not support any form of multitasking operation
- D. It is hard to maintain because parts are tightly coupled

Answer: D

Q495. How does message passing work in a microkernel OS design?

- A. Services exchange messages through microkernel's IPC
- B. Services call each other like regular function calls
- C. All communication happens directly in shared memory
- D. Services communicate through shared global variables

Answer: A

Q496. What is the parameter table approach for passing system call arguments?

- A. It limits system calls to only three parameters total
- B. It stores system call results on the hard disk drive
- C. Parameters are stored in a table; address passed in register
- D. It eliminates the need for any CPU registers entirely

Answer: C

Q497. What advantage does a microkernel have over a monolithic kernel?

- A. Microkernels have more code running inside kernel space
- B. Microkernels always have better raw system performance
- C. Microkernels are easier to extend and port to new hardware
- D. Microkernels eliminate the need for device driver support

Answer: C

Q498. What is the difference between static and dynamic linking?

- A. Dynamic libraries must be recompiled with each application
- B. Static libraries are linked at compile time into binary
- C. Static libraries are always smaller than dynamic libraries
- D. Dynamic libraries are slower but use more total disk space

Answer: B

Q499. Why is debugging a layered OS more straightforward than others?

- A. Because layered systems do not use system calls at all
- B. Because errors in one layer are isolated from other ones
- C. Because all layers share the exact same address space
- D. Because layered systems have no kernel component in them

Answer: B

Q500. What role does the POSIX standard play in operating system design?

- A. It describes physical network topologies for computers
- B. It specifies a standard OS interface for portability use
- C. It defines a mandatory hardware specification for PCs
- D. It is a language used specifically to write OS kernels

Answer: B

Q501. What differs between the short-term and long-term scheduler?

- A. Short-term selects from ready queue; long-term from disk
- B. Long-term runs more frequently than short-term scheduler
- C. Short-term controls the degree of multiprogramming level
- D. They are the same scheduler with two different OS names

Answer: A

Q502. What is a zombie process and why does it occur?

- A. A process created but never actually started running
- B. A terminated process whose parent has not called wait
- C. A process paused by the operating system permanently
- D. A process that runs forever consuming all resources

Answer: B

Q503. How does exec() differ from fork() in Unix systems?

- A. Neither affects the process structure in any way here
- B. exec() creates a new process; fork() replaces image
- C. Both create new processes in exactly identical fashion
- D. fork() creates new process; exec() replaces the image

Answer: D

Q504. What is inter-process communication (IPC) used for?

- A. Connecting a computer to external network devices
- B. Compiling multiple source files at the same time here
- C. Managing CPU scheduling algorithms exclusively now
- D. Allowing processes to exchange data and synchronize

Answer: D

Q505. How do shared memory and message passing IPC differ?

- A. Neither mechanism allows processes to exchange any data
- B. Both mechanisms work identically in all systems always
- C. Message passing uses shared region; shared uses kernel
- D. Shared memory uses common region; messages use kernel

Answer: D

Q506. What is the purpose of the medium-term scheduler?

- A. It schedules I/O operations for device driver code
- B. It manages the file system directory tree structure
- C. It assigns priorities to all newly created processes
- D. It swaps processes between main memory and the disk

Answer: D

Q507. How does a pipe work as an IPC mechanism in Unix?

- A. It provides unidirectional byte stream between procs
- B. It only works between processes on different machines
- C. It replaces the need for any process synchronization
- D. It provides bidirectional communication always here

Answer: A

Q508. What is a process address space composed of?

- A. Exclusively the operating system kernel code only
- B. Only the stack and no other memory segment data
- C. Text, data, heap, and stack segments combined
- D. Only the program code and nothing else at all

Answer: C

Q509. What is cascading termination in process management?

- A. Restarting a process multiple times before stopping
- B. Terminating all children when parent process exits
- C. Gradually slowing down a process before termination
- D. Saving the process state to disk before termination

Answer: B

Q510. What is the role of the dispatcher in process management?

- A. It gives CPU control to the process the scheduler chose
- B. It manages file system allocation tables on the disk
- C. It handles all network communication between hosts
- D. It creates new processes from executable binaries

Answer: A

Q511. What is the convoy effect in FCFS scheduling?

- A. All processes complete at exactly the same time
- B. Short processes wait behind a long CPU-bound one
- C. The CPU is idle while waiting for process arrival
- D. Processes are sorted by priority before execution

Answer: B

Q512. How does the Shortest Remaining Time First algorithm work?

- A. It preempts if a new process has shorter remaining time
- B. It always runs the longest process first for throughput
- C. It is identical to non-preemptive SJF in every way
- D. It uses Round Robin with variable time quantum sizes

Answer: A

Q513. What problem does aging solve in priority scheduling?

- A. It prevents context switching between processes now
- B. It increases the time quantum in Round Robin schemes
- C. It solves starvation for low-priority waiting processes
- D. It eliminates the need for any scheduling algorithm

Answer: C

Q514. What happens with a very small time quantum in Round Robin?

- A. It reduces context switches significantly overall
- B. It increases context switch overhead and degrades speed
- C. It eliminates the need for a ready queue in the system
- D. It converts Round Robin into FCFS scheduling behavior

Answer: B

Q515. How does multilevel queue scheduling organize processes?

- A. Each process gets its own dedicated scheduling queue
- B. Processes are divided into classes with separate queues
- C. All processes share a single common ready queue only
- D. Processes are randomly assigned to available CPU cores

Answer: B

Q516. What is dispatch latency in CPU scheduling?

- A. Time to transfer data from main memory to the CPU
- B. Time to stop one process and start running another
- C. Time for a process to complete its entire execution
- D. Time between submitting a job and its first output

Answer: B

Q517. What is response time and why is it important?

- A. It counts total number of context switches performed
- B. It is time from submission to first response produced
- C. It is the total time to complete the process entirely
- D. It measures amount of memory a process consumes now

Answer: B

Q518. How does SJF achieve optimal average waiting time?

- A. By using a large time quantum to minimize switching
- B. By assigning equal CPU time to every queued process
- C. By running the longest jobs first to clear the queue
- D. By placing shortest jobs first reducing total wait time

Answer: D

Q519. What distinguishes multilevel feedback queue from multilevel queue?

- A. Multilevel queues allow movement but feedback does not
- B. Feedback queues allow processes to move between queues
- C. Both scheduling algorithms are identical in all behavior
- D. Feedback queues do not allow preemption of processes

Answer: B

Q520. What is the main disadvantage of Shortest Job First?

- A. It needs CPU burst time which is hard to predict ahead
- B. It cannot work with more than two processes at a time
- C. It always has the worst average waiting time possible
- D. It causes excessive context switching between processes

Answer: A

Q521. What is the many-to-one threading model?

- A. Many user threads map to one kernel thread
- B. Each user thread maps to one kernel thread
- C. Many kernel threads map to one user thread
- D. Multiple kernel threads share one CPU core

Answer: A

Q522. What is the one-to-one threading model's main advantage?

- A. It eliminates the need for any synchronization at all
- B. It uses less memory than all other threading models
- C. It allows true concurrency on multiprocessor systems
- D. It does not need any operating system kernel support

Answer: C

Q523. What is the purpose of thread pools in multithreaded systems?

- A. To pre-create threads and reuse them for new requests
- B. To prevent any thread from being created in a system
- C. To merge all threads into a single execution context
- D. To permanently store thread objects on the hard disk

Answer: A

Q524. What is thread-local storage (TLS) used for?

- A. Storing thread data on a remote network file server
- B. Sharing data between all threads in a process
- C. Encrypting data that threads access in shared memory
- D. Giving each thread its own copy of specific data

Answer: D

Q525. How does the many-to-many threading model work?

- A. It eliminates both user and kernel threads entirely
- B. It maps one user thread to many kernel threads here
- C. It maps many kernel threads to one user thread now
- D. It maps many user threads to many kernel threads

Answer: D

Q526. What is the signal handling challenge in multithreaded programs?

- A. Signals are only relevant to single-threaded programs
- B. Deciding which thread should receive and handle signal
- C. Signals cannot be generated in multithreaded programs
- D. All signals are handled automatically by the OS kernel

Answer: B

Q527. What is the difference between implicit and explicit threading?

- A. Both approaches are identical in their implementation
- B. Implicit threading needs manual thread creation code
- C. Explicit threading is handled automatically by system
- D. Implicit threading is done by compilers and libraries

Answer: D

Q528. What happens during deferred thread cancellation?

- A. Thread cancellation is impossible in deferred mode here
- B. The target thread is terminated immediately on request
- C. Thread checks for cancellation at cancellation points
- D. The entire process is terminated along with all threads

Answer: C

Q529. Why is fork() complex in multithreaded programs?

- A. fork() terminates the calling thread before the child
- B. It must decide whether to duplicate all or one thread
- C. fork() cannot be called from multithreaded programs
- D. fork() always duplicates all threads without exception

Answer: B

Q530. What is OpenMP and how does it support threading?

- A. OpenMP is a hardware spec for multicore CPUs only
- B. OpenMP is a network protocol for distributed systems
- C. OpenMP provides compiler directives for parallel code
- D. OpenMP is an operating system for embedded controllers

Answer: C

Q531. What are the three requirements for a correct critical section solution?

- A. Speed, memory efficiency, and low power consumption
- B. Simplicity, readability, and backward compatibility now
- C. Atomicity, consistency, isolation, and full durability
- D. Mutual exclusion, progress, and bounded waiting time

Answer: D

Q532. What is the producer-consumer problem in synchronization?

- A. A technique for optimizing database query performance
- B. Coordinating producer adding and consumer removing
- C. A problem about consumer electronics device drivers
- D. A problem about manufacturing physical hardware

Answer: B

Q533. How does Peterson's solution achieve mutual exclusion?

- A. It uses a centralized server to manage all locks
- B. It uses two shared variables: turn and flag array
- C. It uses hardware test-and-set atomic instructions
- D. It relies on disabling all interrupts on processor

Answer: B

Q534. How do spinlocks differ from blocking locks?

- A. Both types always put waiting threads to sleep now
- B. Spinlocks sleep; blocking locks use busy waiting
- C. Both types always use busy waiting to check locks
- D. Spinlocks busy-wait; blocking locks put threads asleep

Answer: D

Q535. What is a monitor in concurrent programming?

- A. A display screen for viewing program output
- B. A network device monitoring all traffic on LAN
- C. A high-level synchronization construct with mutex
- D. A tool for monitoring CPU temperature and speed

Answer: C

Q536. What is the purpose of condition variables in monitors?

- A. They store values of conditional expressions only
- B. They replace the need for any mutual exclusion now
- C. They count threads inside the monitor at any time
- D. They let threads wait for specific conditions safely

Answer: D

Q537. What is the readers-writers problem in synchronization?

- A. A problem about reading user input from a keyboard
- B. Allowing concurrent reads but exclusive writes to data
- C. A technique for writing data to multiple disk drives
- D. A problem about reading and writing files in order

Answer: B

Q538. What is the test-and-set instruction used for?

- A. Testing network and setting bandwidth limits on it
- B. Setting CPU clock speed after testing the hardware
- C. Testing programs and setting breakpoints in them
- D. Atomically testing and modifying a word for locks

Answer: D

Q539. What is the compare-and-swap (CAS) instruction?

- A. A sorting algorithm comparing and swapping values
- B. A technique to compare files and swap if different
- C. A method to swap two processes in scheduling queue
- D. Atomic instruction conditionally updating memory word

Answer: D

Q540. What is a barrier synchronization mechanism?

- A. A way to permanently block a thread from running
- B. A point where all threads must arrive before any go
- C. A physical barrier between two system processors
- D. A mechanism preventing threads from being created

Answer: B

Q541. How does the Banker's algorithm determine if a state is safe?

- A. It finds a safe sequence where all processes complete
- B. It only checks if CPU utilization is above fifty percent
- C. It checks if total resources exceed process count
- D. It verifies no process has any resource allocated now

Answer: A

Q542. How can the hold-and-wait condition be prevented?

- A. By removing mutual exclusion from all shared resources
- B. By giving every process all existing system resources
- C. By requiring processes to request all resources at once
- D. By allowing processes to request at any time freely

Answer: C

Q543. How do deadlock prevention and avoidance differ?

- A. Prevention ignores deadlocks; avoidance terminates all tasks
- B. Prevention detects deadlocks; avoidance recovers from them
- C. Prevention constrains requests; avoidance checks each allocation
- D. They are exactly the same strategy with different names now

Answer: C

Q544. How does total ordering on resources prevent circular wait?

- A. It allows processes to request resources in any order
- B. It makes all resources shareable between all processes
- C. It eliminates the need for any resource allocation logic
- D. Processes must request resources in increasing order now

Answer: D

Q545. What is deadlock detection using a wait-for graph?

- A. A graph showing CPU utilization over time periods
- B. A diagram of physical network connections between PCs
- C. A chart showing memory usage of each running process
- D. A variant with only processes; a cycle means deadlock

Answer: D

Q546. What are the recovery options after detecting a deadlock?

- A. Only restarting the entire computer system works now
- B. Process termination or resource preemption can be used
- C. Deadlocks always resolve themselves automatically here
- D. Adding more physical memory always resolves all deadlock

Answer: B

Q547. What is the ostrich algorithm for handling deadlocks?

- A. A sophisticated algorithm detecting all deadlocks fast
- B. An algorithm preventing deadlock with resource ordering
- C. It ignores the problem assuming deadlocks rarely occur
- D. A technique recovering from deadlocks automatically now

Answer: C

Q548. What is a livelock and how does it differ from deadlock?

- A. Livelock means all processes are blocked and inactive
- B. In livelock processes are active but make no progress
- C. Livelock only occurs in single-threaded applications
- D. Livelock is identical to deadlock in every way here

Answer: B

Q549. What does the Banker's algorithm need from processes?

- A. The total disk space used by each process on system
- B. Only the current CPU usage of each active process
- C. Maximum resource needs declared by each process first
- D. Only the priority level assigned to each process now

Answer: C

Q550. Why might deadlock prevention lead to low resource utilization?

- A. Because conservative restrictions leave resources idle now
- B. Because it allocates all resources to one process always
- C. Because it detects deadlocks only after they have formed
- D. Because it allows processes to hold resources indefinitely

Answer: A

Q551. How does the LRU page replacement algorithm work?

- A. It replaces the page not used for the longest time
- B. It randomly selects any page for replacement now
- C. It replaces the page used furthest in the future
- D. It replaces pages in the order they were loaded in

Answer: A

Q552. What is the Translation Lookaside Buffer (TLB)?

- A. A register storing the base address of process
- B. A buffer for storing data being written to disk
- C. A queue for processes waiting for memory alloc
- D. A fast cache for recent page table entries only

Answer: D

Q553. What is Belady's anomaly in page replacement?

- A. Page faults are constant regardless of the frame count
- B. More frames always reduce the number of page faults
- C. Belady's anomaly only occurs with the optimal algorithm
- D. More frames can increase page faults with FIFO scheme

Answer: D

Q554. How does segmentation differ from paging?

- A. Neither technique uses any address translation now
- B. Both use exactly the same fixed-size memory blocks
- C. Segmentation uses fixed-size; paging uses variable
- D. Paging uses fixed-size; segmentation uses variable

Answer: D

Q555. What is demand paging in virtual memory?

- A. Loading all pages into memory at startup
- B. Requiring users to manually load needed pages
- C. Pre-loading all pages from disk before start
- D. Loading pages only when they are accessed now

Answer: D

Q556. What is the working set model in virtual memory?

- A. Pages a process actively uses in time window
- B. Set of all pages stored on the hard disk now
- C. Total physical memory available in computer
- D. Total number of processes in ready queue

Answer: A

Q557. What is thrashing in virtual memory systems?

- A. A process running very fast and finishing quickly
- B. The process of defragmenting virtual memory space
- C. A technique to improve memory utilization efficiency
- D. Excessive page faults causing most time spent swapping

Answer: D

Q558. How does the FIFO page replacement algorithm work?

- A. It replaces the page with fewest access counts
- B. It replaces pages by their physical address now
- C. It replaces the most recently loaded page here
- D. It replaces the oldest page that was loaded in

Answer: D

Q559. What is the dirty bit used for in paging?

- A. It marks pages containing errors or corrupted data
- B. It indicates whether a page was modified in RAM
- C. It shows whether a page is currently locked in RAM
- D. It counts the number of times a page was accessed

Answer: B

Q560. What is a multi-level page table and why is it used?

- A. A page table stored on multiple different disk drives
- B. A hierarchical page table reducing memory overhead
- C. A page table storing entries for multiple processes
- D. A table mapping pages to multiple physical frames

Answer: B

Q561. How does indexed allocation manage file disk blocks?

- A. Blocks are linked using pointers in each block
- B. An index block contains pointers to data blocks
- C. Blocks are allocated randomly without tracking
- D. All blocks are stored consecutively on disk

Answer: B

Q562. What differs between a hard link and a symbolic link?

- A. Hard links work across file systems but soft do not
- B. Hard link points to inode; soft link points to path
- C. Both hard and symbolic links are exactly identical
- D. Symbolic links point to inodes; hard links to paths

Answer: B

Q563. What is the inode structure in Unix file systems?

- A. A structure storing file metadata and block pointers
- B. A network node handling file transfer protocols
- C. A hardware component managing disk read operations
- D. A process monitoring file system integrity at boot

Answer: A

Q564. How does the FAT file system organize file allocation?

- A. Using a file allocation table chaining disk blocks
- B. Using B-tree indexes for each file on the disk now
- C. Using extent-based allocation with balanced trees
- D. Using inodes and indirect blocks like Unix systems

Answer: A

Q565. What is journaling in modern file systems?

- A. Recording user activity logs for security audit
- B. Writing metadata changes to a log before commit
- C. Storing journal entries for file content searching
- D. Maintaining daily backup schedule of all files now

Answer: B

Q566. What is the free-space management bitmap method?

- A. A method to compress free space for more storage
- B. A technique to visualize the directory tree layout
- C. Each bit represents a block allocation status bit
- D. A graphical representation of files on screen

Answer: C

Q567. What is direct access file organization?

- A. Any record can be accessed directly by position
- B. Files can only be read sequentially from start
- C. Files are accessed by searching all records first
- D. Files can only be accessed through symbolic links

Answer: A

Q568. What is the purpose of the Virtual File System layer?

- A. It provides uniform interface to file system types
- B. It creates virtual copies of files for backup
- C. It manages virtual memory not file system at all
- D. It encrypts all files stored on virtual disk drives

Answer: A

Q569. What is file system consistency checking used for?

- A. Ensuring all files have correct extensions set here
- B. Checking if files contain valid program source
- C. Verifying integrity and repairing corruption now
- D. Checking if file system has enough free space left

Answer: C

Q570. How does contiguous allocation organize file storage?

- A. Files are indexed using a multi-level tree structure
- B. Each file occupies contiguous disk blocks together
- C. Files are in random blocks scattered across the disk
- D. Files are stored using linked list of distant blocks

Answer: B

Q571. How does the SCAN disk scheduling algorithm work?

- A. Head moves one direction servicing then reverses
- B. It randomly selects the next request from queue
- C. It services requests in arrival order always
- D. It always jumps to the closest request first now

Answer: A

Q572. How does SSTF scheduling differ from FCFS?

- A. SSTF services requests in the exact arrival order
- B. SSTF always moves the head to outermost track first
- C. SSTF and FCFS produce identical scheduling results
- D. SSTF services the request nearest to current head

Answer: D

Q573. What is the difference between RAID 0 and RAID 1?

- A. RAID 0 provides redundancy; RAID 1 stripes only
- B. RAID 0 stripes for speed; RAID 1 mirrors for safety
- C. Both provide identical performance and safety here
- D. RAID 0 uses parity; RAID 1 uses error correction

Answer: B

Q574. What is the C-SCAN disk scheduling algorithm?

- A. It services the closest request to minimize seeking
- B. It processes requests in the order submitted always
- C. It scans in both directions alternately each time
- D. Head services one direction then jumps back to start

Answer: D

Q575. What is the purpose of disk defragmentation for HDDs?

- A. Encrypting all data stored on the hard disk drive
- B. Removing all deleted files permanently from disk
- C. Reorganizing fragmented files into contiguous blocks
- D. Increasing physical speed of disk platter rotation

Answer: C

Q576. How does RAID 5 provide fault tolerance?

- A. It keeps three copies of every data block on disk
- B. It distributes parity information across all drives
- C. It mirrors all data on two separate disk drives
- D. It uses error-correcting codes on a separate disk

Answer: B

Q577. Why is SSTF disk scheduling prone to starvation?

- A. Far requests may wait indefinitely for any service
- B. Because it services all requests equally in order
- C. It services farthest request first causing slow times
- D. SSTF scheduling never causes starvation of requests

Answer: A

Q578. What is the disk controller's role?

- A. It manages CPU scheduling for all running processes
- B. It handles network routing for file sharing on LAN
- C. It controls user interface for file management
- D. It translates logical addresses to physical locations

Answer: D

Q579. What is wear leveling in solid-state drives?

- A. Technique to physically level SSD circuit boards
- B. Distributing writes evenly to extend cell lifespan
- C. A process to cool down overheated memory chips now
- D. A method to reduce power consumption of SSD drive

Answer: B

Q580. What is the boot block and where is it stored?

- A. It stores the file system journal for crash recovery
- B. It stores user data in the first track of disk
- C. It contains bootstrap code in a fixed disk location
- D. It contains disk encryption keys for secure access

Answer: C

Q581. How do memory-mapped and port-mapped I/O differ?

- A. Memory-mapped requires special CPU instructions for access
- B. Port-mapped I/O is faster than memory-mapped in every case
- C. Both methods use exactly the same addressing mechanism now
- D. Memory-mapped uses address space; port-mapped uses separate

Answer: D

Q582. How does double buffering improve I/O performance?

- A. One buffer fills while other is being processed by CPU
- B. It eliminates the need for any buffering in the system
- C. It slows down I/O by requiring twice the memory space
- D. It doubles the speed of the physical I/O device hardware

Answer: A

Q583. What is device-independent I/O software?

- A. It handles only network device communication protocol
- B. It makes all devices behave like character devices
- C. It eliminates the need for any device drivers at all
- D. It provides uniform interfacing for diverse devices

Answer: D

Q584. What is an interrupt vector table used for?

- A. Storing data to be written to I/O devices later
- B. Tracking which I/O devices are powered on currently
- C. Mapping interrupt numbers to handler routines fast
- D. Counting total number of interrupts since boot time

Answer: C

Q585. What is the role of the I/O scheduler?

- A. It schedules manufacturing of I/O device hardware
- B. It orders and merges I/O requests for optimal access
- C. It schedules network packets for wire transmission
- D. It determines which user can log into the system now

Answer: B

Q586. How does spooling differ from buffering?

- A. Spooling uses disk as buffer for slow device queues
- B. Buffering uses disk while spooling uses only main memory
- C. Spooling is only used for input devices like keyboards
- D. Spooling and buffering are the same I/O technique here

Answer: A

Q587. What is the device status table?

- A. Storing serial numbers of all connected devices here
- B. Maintaining firmware update history for every device
- C. Displaying device information to end user on screen
- D. Tracking status and request queue for each device now

Answer: D

Q588. How do blocking and non-blocking I/O differ?

- A. Blocking suspends process; non-blocking returns immediately
- B. Both blocking and non-blocking I/O work the same way
- C. Blocking I/O is always faster than non-blocking I/O now
- D. Non-blocking suspends process until I/O completes fully

Answer: A

Q589. What is the DMA controller's role during transfer?

- A. It performs all CPU computations during the transfer
- B. It compiles programs while data is being transferred
- C. It encrypts all data before transferring it to memory
- D. It manages data transfer between device and memory now

Answer: D

Q590. How does asynchronous I/O differ from non-blocking?

- A. Asynchronous I/O is identical to non-blocking I/O now
- B. Non-blocking sends completion notification to caller
- C. Asynchronous notifies process upon I/O completion later
- D. Asynchronous blocks process until data is fully ready

Answer: C

Q591. What is a buffer overflow attack?

- A. It fills printer buffer with too many print jobs
- B. It fills the disk buffer causing file system crash
- C. It overwrites memory beyond buffer to inject code
- D. It overflows network buffer causing packet loss now

Answer: C

Q592. What is role-based access control (RBAC)?

- A. Assigning permissions to roles then roles to users
- B. Assigning permissions directly to individual users
- C. Assigning all permissions to every user in system
- D. Removing all access control from the OS entirely

Answer: A

Q593. How does the access matrix represent protection?

- A. Rows are resources and columns are users only now
- B. It is a physical matrix of hardware protection now
- C. Rows are domains, columns are objects with rights
- D. It only represents network access permissions here

Answer: C

Q594. What is a denial-of-service attack?

- A. Deleting all files from the target system forever
- B. Denying users permission to install software
- C. Encrypting data and demanding ransom for the key
- D. Overwhelming system with requests to make unavailable

Answer: D

Q595. What is a rootkit and why is it dangerous?

- A. Malware hiding itself providing persistent root access
- B. A security scanner checking for root vulnerabilities
- C. A toolkit for administrators to manage root access
- D. A legitimate tool for recovering forgotten passwords

Answer: A

Q596. How does sandboxing improve application security?

- A. It isolates apps in restricted execution environments
- B. It prevents applications from being installed at all
- C. It buries security threats deep in system storage
- D. It encrypts all data processed by sandboxed apps now

Answer: A

Q597. What is two-factor authentication (2FA)?

- A. Using same password on two different computers
- B. Logging into two different accounts at same time
- C. Using two different passwords for one account
- D. Requiring two independent verification for access

Answer: D

Q598. What is an intrusion detection system (IDS)?

- A. Installing security patches on system software now
- B. Managing user passwords and auth credentials only
- C. Monitoring system activity to detect malicious acts
- D. Detecting physical break-ins at office buildings

Answer: C

Q599. How do symmetric and asymmetric encryption differ?

- A. Symmetric uses two keys; asymmetric uses one shared
- B. Both encryption types use the same key mechanism now
- C. Symmetric encryption is always weaker than asymmetric
- D. Symmetric uses one shared key; asymmetric a key pair

Answer: D

Q600. What is privilege escalation in system security?

- A. Exploiting flaws to gain higher privileges than allowed
- B. Reducing user privileges for security enforcement now
- C. Escalating a security ticket to management team here
- D. Legitimately granting more permissions by an admin

Answer: A

Q601. What is the difference between a hard real-time system and a soft real-time system?

- A. Hard real-time guarantees deadlines while soft real-time allows occasional misses
- B. Hard real-time uses hardware timers while soft real-time uses software timers
- C. Hard real-time supports more users than soft real-time
- D. Hard real-time is faster than soft real-time in all scenarios

Answer: A

Q602. How does a multiprocessor system differ from a multicomputer system?

- A. Multiprocessor shares memory while multicomputer has distributed memory
- B. Multiprocessor uses one OS while multicomputer uses multiple keyboards
- C. Multiprocessor is slower than multicomputer in all workloads
- D. Multiprocessor only runs batch jobs while multicomputer handles interactive tasks

Answer: A

Q603. What is the role of a timer interrupt in operating system design?

- A. To prevent user processes from monopolizing the CPU indefinitely
- B. To speed up arithmetic calculations
- C. To increase the amount of available memory
- D. To manage network packet routing

Answer: A

Q604. What is the purpose of privileged instructions in a CPU?

- A. To restrict certain dangerous operations to kernel mode only
- B. To make programs execute faster in user mode
- C. To allow users to directly access I/O devices
- D. To enable parallel processing on multiple cores

Answer: A

Q605. How does symmetric multiprocessing (SMP) differ from asymmetric multiprocessing?

- A. SMP treats all processors equally while asymmetric assigns a master processor
- B. SMP uses one processor while asymmetric uses many
- C. SMP is only for real-time systems while asymmetric is for batch systems
- D. SMP requires identical software on each processor while asymmetric does not

Answer: A

Q606. What is the purpose of a system call in the operating system?

- A. To provide an interface for user programs to request OS kernel services
- B. To compile high-level code into machine language
- C. To physically connect hardware devices to the motherboard
- D. To partition the hard disk into volumes

Answer: A

Q607. What does Direct Memory Access (DMA) accomplish in a computer system?

- A. Allows devices to transfer data to memory without CPU intervention for each byte
- B. Directly increases the size of physical memory
- C. Enables memory to execute instructions without the CPU
- D. Provides encryption for data stored in memory

Answer: A

Q608. What is the memory hierarchy in a computer system?

- A. Arrangement of storage from fast/small registers to slow/large secondary storage
- B. A method for encrypting data in memory
- C. The process of allocating memory to user processes
- D. A technique for compressing files on disk

Answer: A

Q609. What is the purpose of an interrupt vector in an operating system?

- A. To store addresses of interrupt service routines for each interrupt type
- B. To count the number of interrupts that have occurred
- C. To prioritize user processes over system processes
- D. To buffer data during I/O transfers

Answer: A

Q610. What does it mean for an operating system to provide abstraction?

- A. It hides complex hardware details behind simpler interfaces
- B. It makes the hardware run at faster clock speeds
- C. It reduces the physical size of hardware components
- D. It eliminates the need for device drivers entirely

Answer: A

Q611. What is the main trade-off between monolithic and microkernel architectures?

- A. Monolithic has better performance but less modularity than microkernel
- B. Monolithic uses less memory than microkernel in all cases
- C. Microkernel is always faster because it has a smaller kernel
- D. Monolithic supports more hardware devices than microkernel

Answer: A

Q612. How are parameters passed to system calls via registers?

- A. Parameter values are placed directly in CPU registers before the system call
- B. Parameters are written to a file that the kernel reads
- C. Parameters are encoded in the system call number itself
- D. Parameters are passed through network sockets

Answer: A

Q613. What is the key benefit of loadable kernel modules over a static monolithic kernel?

- A. New functionality can be added at runtime without recompiling the kernel
- B. They eliminate the need for device drivers entirely
- C. They make the kernel run in user space
- D. They remove the need for system calls

Answer: A

Q614. What is the run-time environment (RTE) in the context of system calls?

- A. A software layer that intercepts API calls and invokes the correct system calls
- B. A special hardware unit that executes system calls
- C. A debugging tool for testing operating system code
- D. A network protocol for remote procedure calls

Answer: A

Q615. Why is inter-process communication slower in microkernel systems?

- A. Services communicate via message passing through the kernel rather than direct calls
- B. The microkernel uses slower processors
- C. User space runs at reduced clock speed
- D. Microkernel systems cannot use DMA for I/O

Answer: A

Q616. What is the purpose of the /proc file system in Linux?

- A. It provides a virtual file interface to kernel data structures and process info
- B. It stores user documents and personal files
- C. It contains compiled application programs
- D. It holds backup copies of the boot loader

Answer: A

Q617. What is the difference between a policy and a mechanism in OS design?

- A. Mechanism defines how to do something, policy decides what should be done
- B. Mechanism is software while policy is hardware
- C. Policy runs in kernel mode while mechanism runs in user mode
- D. There is no practical difference between the two

Answer: A

Q618. How does the stack-based method of passing system call parameters work?

- A. Parameters are pushed onto the program stack and the kernel pops them off
- B. Parameters are stored in a special CPU cache
- C. Parameters are passed as part of the instruction opcode
- D. Parameters are sent via interrupt signals

Answer: A

Q619. What are information maintenance system calls used for?

- A. Transferring information between the user program and the OS
- B. Creating new user accounts
- C. Formatting storage devices
- D. Establishing network connections

Answer: A

Q620. What advantage does a hybrid kernel offer over a pure microkernel?

- A. It keeps performance-critical services in kernel space while maintaining some modularity
- B. It completely eliminates context switches
- C. It runs all services in user space for maximum reliability
- D. It requires no device drivers

Answer: A

Q621. What is the difference between a process and a program?

- A. A process is an active instance of a program in execution with its own state
- B. A program runs on the CPU while a process is stored on disk
- C. A process is always larger in size than a program
- D. There is no difference between a process and a program

Answer: A

Q622. What is the purpose of the process address space?

- A. It defines the memory layout including text, data, heap, and stack sections
- B. It specifies the physical location of the process on the hard disk
- C. It determines the network address used by the process
- D. It sets the display resolution for the process window

Answer: A

Q623. How does message passing IPC differ from shared memory IPC in terms of synchronization?

- A. Message passing provides implicit synchronization while shared memory requires explicit synchronization
- B. Shared memory is always faster for large data transfers
- C. Message passing cannot be used between processes on the same machine
- D. Shared memory does not require any OS support

Answer: A

Q624. What is a message queue as an IPC mechanism?

- A. A linked list of messages stored in the kernel that processes can read and write
- B. A hardware buffer on the network card
- C. A queue of CPU instructions waiting to execute
- D. A list of print jobs waiting for the printer

Answer: A

Q625. What is the difference between direct and indirect inter-process communication?

- A. Direct IPC names the recipient process explicitly while indirect uses shared mailboxes
- B. Direct IPC is faster because it skips the kernel
- C. Indirect IPC can only be used in distributed systems
- D. Direct IPC requires shared memory while indirect does not

Answer: A

Q626. What information is saved during a context switch?

- A. The CPU registers, program counter, and process state of the outgoing process
- B. Only the process name and user ID
- C. The contents of the hard disk
- D. The screen display buffer

Answer: A

Q627. What is the role of the job queue in process scheduling?

- A. It contains all processes in the system regardless of their state
- B. It only holds processes currently executing on the CPU
- C. It stores terminated processes waiting for cleanup
- D. It manages disk I/O requests

Answer: A

Q628. How does the Unix kill() system call work?

- A. It sends a specified signal to a process identified by its PID
- B. It immediately destroys the process without any cleanup
- C. It only works on child processes of the calling process
- D. It deletes the executable file of the target process

Answer: A

Q629. What is the difference between I/O-bound and CPU-bound processes?

- A. I/O-bound processes spend more time on I/O operations while CPU-bound spend more time computing
- B. I/O-bound processes are faster than CPU-bound processes
- C. CPU-bound processes use more memory than I/O-bound processes
- D. There is no meaningful difference between them

Answer: A

Q630. What is swapping in the context of process management?

- A. Moving a process between main memory and a backing store to free memory
- B. Exchanging data between two running processes
- C. Switching the CPU between user mode and kernel mode
- D. Replacing one operating system with another

Answer: A

Q631. Why does Shortest Job First (SJF) scheduling produce the minimum average waiting time?

- A. Shorter jobs finish quickly, reducing the waiting time for all subsequent jobs
- B. It uses the fastest CPU clock speed for short jobs
- C. It skips I/O operations to save time
- D. It runs all processes simultaneously

Answer: A

Q632. What is the problem of starvation in priority scheduling?

- A. Low-priority processes may never execute because higher-priority processes keep arriving
- B. All processes starve for memory simultaneously
- C. The CPU runs out of power during execution
- D. Processes consume too much disk space

Answer: A

Q633. How does the aging technique prevent starvation?

- A. It gradually increases the priority of waiting processes over time
- B. It reduces the priority of all running processes
- C. It terminates processes that have waited too long
- D. It allocates more memory to starving processes

Answer: A

Q634. What is the effect of time quantum size on Round Robin performance?

- A. Too large behaves like FCFS; too small causes excessive context switching overhead
- B. Larger quanta always improve performance
- C. Smaller quanta always improve response time without any downside
- D. Time quantum size has no effect on system performance

Answer: A

Q635. In multilevel queue scheduling, how are processes assigned to queues?

- A. Based on process characteristics such as type, priority, or memory requirements
- B. Randomly distributed among available queues
- C. All placed in a single queue sorted by arrival time
- D. Assigned based on the day of the week they were created

Answer: A

Q636. What advantage does multilevel feedback queue scheduling have over multilevel queue?

- A. Processes can move between queues based on their behavior allowing dynamic adaptation
- B. It uses fewer CPU resources for scheduling decisions
- C. It eliminates the need for context switching
- D. It guarantees that all processes finish in the same amount of time

Answer: A

Q637. What is the difference between preemptive SJF and non-preemptive SJF?

- A. Preemptive SJF interrupts a running process if a shorter one arrives; non-preemptive does not
- B. Non-preemptive SJF is always faster than preemptive SJF
- C. Preemptive SJF does not require knowing burst times
- D. Non-preemptive SJF causes more context switches

Answer: A

Q638. What is the role of the medium-term scheduler in CPU scheduling?

- A. It swaps processes in and out of memory to control the degree of multiprogramming
- B. It selects which process runs on the CPU next
- C. It decides which jobs enter the system from the job pool
- D. It manages disk I/O scheduling

Answer: A

Q639. Why is predicting the next CPU burst length important for SJF scheduling?

- A. SJF requires knowing burst lengths to select the shortest job but future bursts are unknown
- B. Burst length determines how much memory a process needs
- C. Longer bursts indicate higher priority processes
- D. Burst prediction is only needed for real-time systems

Answer: A

Q640. What is the difference between throughput and turnaround time as scheduling metrics?

- A. Throughput measures processes completed per time unit while turnaround time measures total time for one process
- B. Throughput is always larger than turnaround time
- C. They measure the same thing using different units
- D. Turnaround time applies only to I/O-bound processes

Answer: A

Q641. What challenge does Amdahl's Law highlight for multicore systems?

- A. Adding more memory always improves performance linearly
- B. The serial portion of a program limits the maximum speedup achievable through parallelism
- C. All programs can be fully parallelized
- D. Multicore systems are slower than single-core systems

Answer: B

Q642. What is the key disadvantage of the one-to-one threading model?

- A. It does not support kernel-level threading
- B. Creating a user thread requires creating a corresponding kernel thread, adding overhead
- C. It cannot run on multiprocessor systems
- D. It merges all threads into a single execution context

Answer: B

Q643. What is the purpose of the `pthread_join()` function?

- A. To create a new thread
- B. To wait for a specific thread to terminate before continuing
- C. To send a signal to another thread
- D. To detach a thread from the process

Answer: B

Q644. What happens when a user-level thread makes a blocking system call in a many-to-one model?

- A. Only that thread blocks while others continue
- B. The entire process blocks because the single kernel thread is blocked
- C. The thread is automatically converted to a kernel thread
- D. The system call is ignored

Answer: B

Q645. What is the difference between task parallelism and data parallelism?

- A. Task parallelism distributes data across cores; data parallelism distributes tasks
- B. Data parallelism distributes subsets of data across cores performing the same operation; task parallelism distributes different tasks across cores
- C. They are different names for the same concept
- D. Data parallelism only works with databases

Answer: B

Q646. What is the purpose of `pthread_cancel()` in the Pthreads API?

- A. To create a cancelable thread
- B. To request termination of a target thread
- C. To pause a thread temporarily
- D. To increase a thread's priority

Answer: B

Q647. Why is asynchronous thread cancellation considered more dangerous than deferred cancellation?

- A. It uses more memory
- B. The target thread is terminated immediately, potentially leaving shared data in an inconsistent state
- C. It requires kernel mode exclusively
- D. It can only cancel one thread at a time

Answer: B

Q648. What advantage does a thread pool provide over creating threads on demand?

- A. Thread pools use more memory
- B. Pre-created threads reduce the overhead of thread creation and limit the total number of threads
- C. Thread pools eliminate the need for synchronization
- D. Thread pools allow only one thread at a time

Answer: B

Q649. In the context of threading, what is the fork-join model?

- A. A model where threads can only fork new processes
- B. A model where a parent thread creates child threads that execute tasks and then join back with the parent
- C. A model specific to database transactions
- D. A model that prevents any thread from terminating

Answer: B

Q650. What is the relationship between threads and signals in UNIX?

- A. Signals can only be sent to processes, never to individual threads
- B. A signal may be delivered to the specific thread to which it applies, to every thread, or to a designated thread
- C. Threads automatically ignore all signals
- D. Each thread has its own independent signal handler table

Answer: B

Q651. How does the test-and-set instruction help solve the critical section problem?

- A. It tests whether the CPU is available
- B. It atomically reads a variable and sets it to true, enabling mutual exclusion without interruption
- C. It sets all variables in memory to zero
- D. It tests the speed of the system bus

Answer: B

Q652. In the bounded buffer problem using semaphores, what are the three semaphores typically used?

- A. read, write, execute
- B. mutex (for mutual exclusion), full (counts filled slots), empty (counts empty slots)
- C. start, stop, pause
- D. lock, unlock, wait

Answer: B

Q653. What is the key challenge in the readers-writers problem?

- A. Ensuring that files can only be read once
- B. Allowing multiple readers simultaneous access while ensuring writers get exclusive access
- C. Preventing any process from reading data
- D. Synchronizing disk rotational speed

Answer: B

Q654. How does priority inversion occur in synchronization?

- A. A high-priority process runs before a low-priority process
- B. A high-priority process is blocked waiting for a lock held by a low-priority process, which is itself preempted by medium-priority processes
- C. All processes run with the same priority
- D. The scheduler inverts the order of the ready queue

Answer: B

Q655. What is the purpose of a condition variable in a monitor?

- A. To store the return value of a function
- B. To allow a thread to wait until a particular condition is met, and to be notified when it changes
- C. To count the number of active monitors
- D. To monitor CPU temperature conditions

Answer: B

Q656. What is the difference between wait() and signal() on a condition variable versus on a semaphore?

- A. They are identical in behavior
- B. Condition variable signal() has no effect if no thread is waiting, while semaphore signal() always increments the value
- C. Semaphore wait() never blocks
- D. Condition variables can only be used in Java

Answer: B

Q657. What problem can occur when semaphores are used incorrectly, such as swapping the order of wait() operations?

- A. Faster execution
- B. Deadlock, where two or more processes are each waiting on semaphores held by the other
- C. Automatic error correction
- D. Improved throughput

Answer: B

Q658. What is a reentrant lock and how does it differ from a standard mutex?

- A. A reentrant lock can only be acquired once
- B. A reentrant lock allows the same thread to acquire it multiple times without deadlocking
- C. A reentrant lock is slower than a standard mutex in all cases
- D. A reentrant lock provides no mutual exclusion

Answer: B

Q659. Why are spinlocks more efficient than blocking locks on multiprocessor systems for short critical sections?

- A. Spinlocks use less memory
- B. Spinning avoids the overhead of context switching, which would take longer than the short wait time
- C. Spinlocks automatically parallelize the critical section
- D. Blocking locks are not supported on multiprocessors

Answer: B

Q660. How does imposing a total ordering on resource types prevent circular wait?

- A. It requires processes to release all resources before requesting new ones
- B. Processes must request resources in increasing order of their assigned numbers, making circular dependencies impossible
- C. It limits the number of resources available
- D. It assigns equal priority to all resources

Answer: B

Q661. In the Banker's Algorithm, what does the 'Available' vector represent?

- A. The total number of resources in the system
- B. The number of instances of each resource type currently available for allocation
- C. The maximum resources any process can request
- D. The resources that have been permanently allocated

Answer: B

Q662. What factors should be considered when selecting a victim process for deadlock recovery?

- A. Only the process name
- B. Priority, computation time completed, resources held, remaining time, and number of rollbacks
- C. Only the process ID number
- D. The color of the user's terminal

Answer: B

Q663. What is the difference between deadlock prevention and deadlock avoidance?

- A. They are identical approaches
- B. Prevention ensures one of the four necessary conditions can never hold; avoidance dynamically checks each request to stay in a safe state
- C. Avoidance eliminates resources; prevention adds resources
- D. Prevention is used only in distributed systems

Answer: B

Q664. What is the safe sequence in the context of the Banker's Algorithm?

- A. A random ordering of all processes
- B. An ordering of processes such that each process can obtain its maximum required resources from currently available resources plus resources held by all preceding processes
- C. The order in which processes were created
- D. A sequence where no process ever requests resources

Answer: B

Q665. How can the hold-and-wait condition be prevented by requiring processes to request all resources at once?

- A. Processes request resources one at a time
- B. A process must request all needed resources before execution begins; it cannot hold some and wait for others
- C. Processes share all resources simultaneously
- D. The OS holds all resources and never allocates them

Answer: B

Q666. What is the difference between process termination and resource preemption as deadlock recovery methods?

- A. They are the same approach
- B. Process termination kills processes to break deadlock; resource preemption forcibly takes resources from processes and reassigns them
- C. Resource preemption always terminates all processes
- D. Process termination only pauses processes temporarily

Answer: B

Q667. Why is the Banker's Algorithm considered impractical for many real systems?

- A. It is too simple to implement
- B. It requires processes to declare their maximum resource needs in advance, which is often unknown
- C. It only works with one resource type
- D. It causes deadlock instead of preventing it

Answer: B

Q668. In deadlock detection with multiple resource instances, what algorithm is used?

- A. Dijkstra's shortest path algorithm
- B. A variant of the Banker's Algorithm that checks if the current state can complete all processes
- C. Binary search on resource allocation
- D. Depth-first search on a binary tree

Answer: B

Q669. What is starvation in the context of deadlock recovery, and how can it be prevented?

- A. Starvation is when the CPU runs out of power
- B. The same process is repeatedly selected as the victim for rollback; prevented by including rollback count in victim selection criteria
- C. Starvation means processes do not have enough memory
- D. It is a type of deadlock that cannot be prevented

Answer: B

Q670. How does a Translation Lookaside Buffer (TLB) improve memory access performance?

- A. By increasing the physical memory size
- B. By caching recently used page table entries to avoid the overhead of accessing the page table in memory for every address translation
- C. By compressing data in memory
- D. By reducing the CPU clock speed

Answer: B

Q671. What is the effective memory access time if the TLB hit ratio is 90%, TLB access time is 10ns, and memory access time is 100ns?

- A. 100ns
- B. 120ns
- C. 110ns
- D. 200ns

Answer: B

Q672. What is the advantage of a multi-level page table over a single-level page table?

- A. It is faster to look up
- B. It reduces the memory required for the page table by only allocating entries for used portions of the address space
- C. It eliminates the need for a TLB
- D. It increases the page size

Answer: B

Q673. How does the optimal (OPT) page replacement algorithm work?

- A. It replaces the most recently used page
- B. It replaces the page that will not be used for the longest period of time in the future
- C. It replaces pages in FIFO order
- D. It replaces the page with the lowest page number

Answer: B

Q674. Why does the FIFO page replacement algorithm suffer from Belady's anomaly?

- A. Because it uses too much memory
- B. Increasing the number of frames can lead to more page faults because FIFO does not consider page usage frequency
- C. Because it always replaces the most recently used page
- D. Because it requires future knowledge of page references

Answer: B

Q675. What is the working set of a process?

- A. The total number of pages in the process
- B. The set of pages that the process has referenced within a recent time window
- C. The set of all physical frames in the system
- D. The set of processes currently in the ready queue

Answer: B

Q676. How does segmentation differ from paging in memory management?

- A. Segmentation uses fixed-size blocks; paging uses variable-size blocks
- B. Segmentation divides memory into variable-size logical units (code, data, stack); paging divides into fixed-size physical blocks
- C. Paging is visible to the programmer; segmentation is not
- D. They are identical approaches with different names

Answer: B

Q677. What is the purpose of the dirty (modified) bit in a page table entry?

- A. To indicate the page contains errors
- B. To indicate the page has been modified since it was loaded, so it must be written back to disk before replacement
- C. To mark the page as deleted
- D. To count how many times the page was accessed

Answer: B

Q678. What causes thrashing and what is its primary symptom?

- A. Too many CPU cores causing overheating
- B. A process spends more time paging (swapping pages in and out) than executing, causing severe performance degradation
- C. The hard disk runs out of space
- D. Too many users are logged in simultaneously

Answer: B

Q679. What is the inverted page table and how does it save memory?

- A. A page table stored upside down in memory
- B. It has one entry per physical frame instead of one per virtual page, reducing table size for large address spaces
- C. It inverts the logical-to-physical mapping to physical-to-logical
- D. It is a page table that grows dynamically

Answer: B

Q680. How does linked allocation manage disk blocks for a file?

- A. All blocks are stored contiguously on disk
- B. Each block contains a pointer to the next block, forming a linked list of disk blocks
- C. A central index block stores pointers to all data blocks
- D. Blocks are stored in a hash table

Answer: B

Q681. What is the advantage of indexed allocation over linked allocation?

- A. It uses less disk space per file
- B. It supports direct (random) access by storing all block pointers in a single index block
- C. It never wastes any disk space
- D. It is simpler to implement

Answer: B

Q682. What is journaling in file systems and what problem does it solve?

- A. It keeps a human-readable log of user activity
- B. It records file system changes in a log before committing them, enabling recovery from crashes without full consistency checks
- C. It creates daily backups of all files
- D. It journals CPU usage for each file

Answer: B

Q683. How does the bitmap (bit vector) method manage free disk space?

- A. It stores free blocks in a linked list
- B. Each bit represents a disk block — 1 for free, 0 for allocated (or vice versa), allowing quick identification of free blocks
- C. It counts the total number of free blocks without tracking which ones
- D. It maps file names to free blocks

Answer: B

Q684. What is the difference between a hard link and a symbolic link in terms of inode behavior?

- A. Both create a new inode for the linked file
- B. A hard link creates another directory entry pointing to the same inode; a symbolic link creates a new inode containing the path to the target
- C. A symbolic link shares the same inode as the original file
- D. Hard links work across different file systems

Answer: B

Q685. What does the Virtual File System (VFS) layer provide?

- A. A virtual disk for storage
- B. A common interface that allows the OS to support multiple file system types transparently
- C. Virtual memory for file operations
- D. Encryption for all file systems

Answer: B

Q686. What is an Access Control List (ACL) in the context of file systems?

- A. A list of all files in a directory
- B. A per-file list specifying which users or groups have which permissions on that file
- C. A list of CPU access times for files
- D. A list of all mounted file systems

Answer: B

Q687. What is the purpose of the superblock in a Unix file system?

- A. To store user passwords
- B. To store metadata about the file system itself, including its size, block size, free blocks count, and inode information
- C. To store the largest file in the system
- D. To manage network connections

Answer: B

Q688. What is contiguous allocation's main advantage and disadvantage?

- A. Advantage: no fragmentation; Disadvantage: slow access
- B. Advantage: fast sequential and direct access; Disadvantage: external fragmentation and difficulty growing files
- C. Advantage: no wasted space; Disadvantage: complex implementation
- D. Advantage: supports journaling; Disadvantage: requires large inodes

Answer: B

Q689. What is the difference between ordered mode and writeback mode journaling?

- A. They are identical journaling modes
- B. Ordered mode writes data to disk before journaling metadata; writeback mode journals metadata without guaranteeing data order, risking stale data after crash
- C. Writeback mode is always slower than ordered mode
- D. Ordered mode only journals file names

Answer: B

Q690. How does the SCAN (elevator) algorithm work for disk scheduling?

- A. It randomly selects the next request to service
- B. The disk arm moves in one direction servicing requests until it reaches the end, then reverses direction and services requests going back
- C. It services only requests on the outermost tracks
- D. It services requests in alphabetical order of file names

Answer: B

Q691. What advantage does C-SCAN have over SCAN?

- A. C-SCAN is simpler to implement
- B. C-SCAN provides more uniform wait times by only servicing requests in one direction and returning to the beginning without servicing
- C. C-SCAN uses less disk space
- D. C-SCAN eliminates all seek time

Answer: B

Q692. How does RAID 5 distribute parity information?

- A. Parity is stored on a dedicated disk
- B. Parity blocks are distributed across all disks in the array, avoiding the bottleneck of a single parity disk
- C. Parity is not used in RAID 5
- D. Each disk stores its own parity only

Answer: B

Q693. What is wear leveling in SSDs?

- A. Evenly distributing weight on the drive housing
- B. Distributing write operations evenly across all flash memory cells to prevent some cells from wearing out faster than others
- C. Reducing the speed of write operations to extend drive life
- D. Leveling the surface of the flash memory chips

Answer: B

Q694. What is the LOOK variant of disk scheduling?

- A. The disk arm looks ahead one request at a time
- B. The arm moves in one direction only as far as the last pending request, then reverses, instead of going to the disk's end
- C. It looks at all requests simultaneously before choosing
- D. It skips every other request

Answer: B

Q695. What is the boot block on a disk?

- A. A block that prevents booting
- B. The disk block(s) containing the bootstrap loader program that initializes the system and loads the operating system
- C. A block reserved for backup data
- D. The largest block on the disk

Answer: B

Q696. What is disk defragmentation and when is it beneficial?

- A. It encrypts disk data for security
- B. It reorganizes scattered file fragments into contiguous blocks, beneficial for HDDs to reduce seek times but unnecessary for SSDs
- C. It deletes duplicate files from the disk
- D. It increases the disk's storage capacity

Answer: B

Q697. What is the purpose of a hot spare disk in a RAID configuration?

- A. A disk that stores temporary data
- B. An idle disk that automatically replaces a failed disk in the array, reducing the window of vulnerability
- C. A disk that runs at a higher temperature
- D. A disk used only for read operations

Answer: B

Q698. What is memory-mapped I/O and how does it differ from port-mapped I/O?

- A. Memory-mapped I/O is slower than port-mapped I/O in all cases
- B. Memory-mapped I/O maps device registers into the regular memory address space; port-mapped I/O uses separate I/O address space with special instructions
- C. Port-mapped I/O uses regular memory addresses
- D. Memory-mapped I/O requires dedicated I/O instructions

Answer: B

Q699. What is double buffering and why is it used?

- A. Using two CPUs for I/O operations
- B. Using two buffers so the device can fill one while the process reads from the other, overlapping I/O and computation
- C. Buffering data twice for error checking
- D. Using buffers of double the normal size

Answer: B

Q700. How does an interrupt vector table work?

- A. It stores all device drivers in a single table
- B. It is an array of pointers to interrupt handler routines, indexed by interrupt number, allowing quick dispatch to the correct handler
- C. It converts analog signals to digital interrupts
- D. It counts the number of interrupts per second

Answer: B

Q701. What is the role of the DMA controller during a data transfer?

- A. It replaces the CPU entirely during the transfer
- B. It manages the transfer of data between I/O devices and memory, handling addressing, counting, and bus arbitration while the CPU runs other tasks
- C. It only monitors the transfer without participating
- D. It encrypts data during transfer

Answer: B

Q702. How does asynchronous I/O differ from non-blocking I/O?

- A. They are identical concepts
- B. Non-blocking I/O returns immediately with available data; asynchronous I/O initiates the operation and notifies the process when it completes, fully decoupling the I/O from the process
- C. Asynchronous I/O blocks the process until completion
- D. Non-blocking I/O uses more memory than asynchronous I/O

Answer: B

Q703. What is the device status table maintained by the OS?

- A. A table listing all devices that need repair
- B. A table that tracks the type, address, status, and pending I/O requests for each device in the system
- C. A table of device prices for inventory management
- D. A table mapping device names to driver versions

Answer: B

Q704. What is device-independent I/O software responsible for?

- A. Manufacturing hardware devices
- B. Providing uniform interfaces, buffering, error reporting, and device allocation that work regardless of the specific device
- C. Installing device drivers automatically
- D. Converting device signals to audio

Answer: B

Q705. What is an interrupt controller and what does it do?

- A. A software program that generates interrupts
- B. Hardware that manages multiple interrupt lines, prioritizes interrupts, and routes them to the CPU
- C. A controller that disables all interrupts permanently
- D. A device that converts interrupts to I/O requests

Answer: B

Q706. What is the access matrix model and how is it used for protection?

- A. A matrix that stores encryption keys
- B. A conceptual model with rows for domains (subjects) and columns for objects, where each cell contains the set of operations the domain can perform on the object
- C. A physical matrix of hardware protection circuits
- D. A matrix that maps memory addresses

Answer: B

Q707. How does Role-Based Access Control (RBAC) simplify security management?

- A. By eliminating all access control
- B. By assigning permissions to roles instead of individual users, then assigning users to roles, reducing the complexity of managing per-user permissions
- C. By giving all users the same role
- D. By using biometrics exclusively

Answer: B

Q708. What is a buffer overflow attack and why is it dangerous?

- A. Filling a printer buffer with too much data
- B. Writing data beyond a buffer's allocated memory to overwrite adjacent memory, potentially hijacking program control flow to execute malicious code
- C. Downloading too many files at once
- D. Overflowing the CPU's instruction cache

Answer: B

Q709. What is sandboxing in operating system security?

- A. Running programs on beach-themed desktops
- B. Running a program in an isolated environment with restricted access to system resources, limiting the damage if the program is compromised
- C. Encrypting program data with sand-based algorithms
- D. A technique to speed up program execution

Answer: B

Q710. What is privilege escalation?

- A. A user requesting a password reset
- B. An attacker exploiting a vulnerability to gain higher access privileges than initially granted, such as obtaining root access from a regular user account
- C. The process of granting administrator access to new employees
- D. Upgrading the operating system to a newer version

Answer: B

Q711. How does the capability list approach implement the access matrix?

- A. It stores one list per object with authorized subjects
- B. Each process holds a list of capabilities (tokens) specifying objects it can access and the permitted operations
- C. It stores a single global list of all permissions
- D. It uses a hash table indexed by user names

Answer: B

Q712. What is a rootkit and why is it particularly dangerous?

- A. A toolkit for system administrators to manage root access
- B. Malware that hides its presence by modifying the OS or firmware, making it extremely difficult to detect while maintaining persistent unauthorized access
- C. A set of root vegetables used as a screen saver
- D. A backup tool for the root partition

Answer: B

Q713. What are protection rings and how do they enforce security?

- A. Physical rings placed around the CPU for protection
- B. Hierarchical privilege levels (Ring 0 being most privileged) enforced by hardware, where inner rings have more privileges and outer rings are restricted
- C. Software-only protection mechanisms in the firewall
- D. Network security zones for different departments

Answer: B

Q714. What is the purpose of a turnstile synchronization mechanism used in some kernels?

- A. To limit the number of threads in the system
- B. A queue structure where blocked threads waiting for a lock are placed, allowing priority inheritance and ordered wakeup
- C. To rotate CPU access among threads
- D. To count the number of active synchronization primitives

Answer: B

Q715. What is the C-LOOK disk scheduling algorithm?

- A. A variant of C-SCAN that reverses at the end of the disk
- B. A variant of C-SCAN where the arm only goes as far as the last pending request in each direction before wrapping around, avoiding unnecessary travel to disk edges
- C. A scheduling algorithm that looks at every other request
- D. An algorithm identical to FCFS but circular

Answer: B

Q716. What is the difference between low-level formatting and high-level formatting of a disk?

- A. They are the same process with different names
- B. Low-level formatting creates the physical sector structure on the disk surface; high-level formatting creates the file system data structures within a partition
- C. High-level formatting erases the physical platters
- D. Low-level formatting only applies to SSDs

Answer: B

Q717. What is the purpose of the I/O request queue in the OS?

- A. To store the output of completed I/O operations
- B. To hold pending I/O requests for a device, allowing the scheduler to reorder them for efficiency
- C. To count the total number of I/O operations performed
- D. To buffer all keyboard inputs

Answer: B

Q718. How does caching differ from buffering in the context of I/O?

- A. They are identical concepts with different names
- B. A buffer holds the only copy of data during transfer; a cache holds a faster duplicate copy of data that also exists elsewhere
- C. Caching is always slower than buffering
- D. Buffering uses CPU cache while caching uses RAM

Answer: B

Q719. What is the purpose of a digital certificate in security?

- A. To compress encrypted data
- B. To bind a public key to an entity's identity, verified by a trusted Certificate Authority, enabling secure key exchange
- C. To digitally sign physical documents
- D. To store all of a user's passwords

Answer: B

Q720. What is a man-in-the-middle (MITM) attack?

- A. A physical attack on a server's hardware
- B. An attacker secretly intercepts and potentially alters communication between two parties who believe they are communicating directly with each other
- C. A type of virus that hides in the middle of a file
- D. An attack that only targets the middle layer of the network stack

Answer: B

Hard Questions

360 questions

Q721. In the context of OS design, what is the key advantage of a microkernel architecture over a monolithic kernel?

- A. Better performance from fewer context switches in kernel
- B. Direct hardware access available for all system services
- C. Higher reliability since most services run in user space
- D. Simpler implementation compared to other kernel designs

Answer: C

Q722. What is the purpose of a hypervisor in virtualization?

- A. To serve as a real-time antivirus scanning tool
- B. To create and manage virtual machines on hardware
- C. To increase the total RAM capacity of a system
- D. To compile user programs at a much faster speed

Answer: B

Q723. What distinguishes a Type 1 hypervisor from a Type 2 hypervisor?

- A. Type 1 runs on top of host OS, Type 2 on hardware
- B. There is no real or meaningful difference between them
- C. Type 1 is inherently slower in performance than Type 2
- D. Type 1 runs directly on hardware, Type 2 on host OS

Answer: D

Q724. In a hard real-time system, what happens when a deadline is missed?

- A. The task priority gets increased
- B. The system gradually slows down
- C. The task is automatically retried
- D. The result is a total system failure

Answer: D

Q725. What is the main purpose of the interrupt vector table?

- A. To hold addresses of interrupt service routines
- B. To schedule and dispatch CPU tasks to run
- C. To manage the overall file system structure
- D. To permanently store user programs on disk

Answer: A

Q726. What is the concept of 'dual-mode operation' in operating systems and why is it essential?

- A. It separates user and kernel modes for protection
- B. It doubles the overall CPU processing speed rate
- C. It enables running two operating systems at once
- D. It allows two users to work at the same time

Answer: A

Q727. In clustered systems, what is the difference between asymmetric clustering and symmetric clustering?

- A. Asymmetric has standby node; symmetric all nodes share
- B. Asymmetric uses different OS, symmetric uses same OS
- C. There is no meaningful difference between these approaches
- D. Asymmetric clustering is inherently faster than symmetric

Answer: A

Q728. What mechanism does the OS use to prevent a user program from running indefinitely and never returning control?

- A. Timer interrupt
- B. Watchdog timer
- C. Busy waiting
- D. Device polling

Answer: A

Q729. What is the purpose of caching in the memory hierarchy of an operating system?

- A. To increase the size of main memory
- B. To store frequent data in faster memory
- C. To permanently store data on disk
- D. To compress data for efficient storage

Answer: B

Q730. What problem does cache coherency address in multiprocessor systems?

- A. Preventing the processor cache from overflowing
- B. Reducing the overall processor cache miss rate
- C. Ensuring all processor caches are the same size
- D. Ensuring shared data updates reflect in all caches

Answer: D

Q731. In the Mach microkernel, what are the primary abstractions provided?

- A. Files, directories, and named pipes
- B. Processes, semaphores, and mutexes
- C. Blocks, inodes, and superblocks
- D. Tasks, threads, ports, and messages

Answer: D

Q732. What is an exokernel?

- A. A kernel that runs outside the main CPU
- B. A minimal kernel giving apps direct hardware control
- C. A kernel designed only for external USB devices
- D. A kernel that runs entirely in user space only

Answer: B

Q733. What is the purpose of a policy/mechanism separation in OS design?

- A. To separate what is done from how it is done
- B. To divide the OS into two equal-sized halves
- C. To separate user programs from system programs
- D. To directly improve hardware processing speed

Answer: A

Q734. What is the main concept behind a unikernel?

- A. A kernel that runs on one processor core only
- B. A kernel that supports only one single user
- C. A kernel composed of one single code module
- D. A single-image OS compiled with the application

Answer: D

Q735. In OS debugging, what is a core dump?

- A. A memory snapshot of a process at crash time
- B. A network diagnostic and monitoring utility
- C. A listing of all currently running processes
- D. A disk formatting and partitioning operation

Answer: A

Q736. What is the strace command used for in Linux?

- A. To trace network packets on the wire
- B. To trace system calls made by a process
- C. To trace file system changes on the disk
- D. To trace memory allocations in the heap

Answer: B

Q737. What is the advantage of the modular kernel approach over a pure layered approach?

- A. It uses significantly less memory than layered design
- B. It is much simpler to implement and maintain
- C. Modules call any other module directly with flexibility
- D. It does not require loading any device drivers at all

Answer: C

Q738. In system generation (SYSGEN), the OS is configured for a specific hardware platform. Which of the following is NOT typically determined during SYSGEN?

- A. CPU type and count
- B. User access passwords
- C. Connected peripherals
- D. Available memory size

Answer: B

Q739. What is the role of the Hardware Abstraction Layer (HAL) in Windows?

- A. To manage and track user login sessions
- B. To provide a graphical user interface
- C. To hide hardware differences for portability
- D. To compile and link device driver programs

Answer: C

Q740. What distinguishes a library OS from a traditional OS design?

- A. Apps link directly with OS library code
- B. It manages library catalog structures
- C. It is a collection of shared libraries
- D. It only manages shared library files

Answer: A

Q741. In Chrome's multi-process architecture, what types of processes does it use?

- A. Only renderer display processes
- B. Background and foreground processes
- C. Only a single monolithic process
- D. Browser, renderer, and plug-in processes

Answer: D

Q742. What is the difference between direct and indirect communication in message passing?

- A. Direct names sender/receiver; indirect uses mailboxes
- B. Direct is synchronous only; indirect is asynchronous
- C. Direct uses mailboxes; indirect names each process
- D. There is no difference between the two

Answer: A

Q743. What is the rendezvous mechanism in synchronous message passing?

- A. A method to forcefully terminate processes
- B. Both sender and receiver block until ready
- C. A strategy for memory allocation
- D. A technique for CPU load balancing

Answer: B

Q744. In UNIX, how does the init process (PID 1) handle orphan processes?

- A. It adopts them and collects their exit status
- B. It moves them to a special waiting queue
- C. It terminates orphan processes immediately
- D. It ignores them and frees their resources

Answer: A

Q745. What is a socket in the context of IPC?

- A. A type of distributed file system
- B. A physical hardware port connector
- C. An endpoint defined by IP and port
- D. A special-purpose CPU register set

Answer: C

Q746. What is Remote Procedure Call (RPC) in the context of distributed process communication?

- A. A file transfer protocol for distributed systems
- B. Calling a remote procedure as if it were local
- C. A method to restart processes on remote machines
- D. A type of CPU scheduling algorithm for clusters

Answer: B

Q747. What is the purpose of the medium-term scheduler?

- A. To select the next process for CPU execution
- B. To create new processes from program images
- C. To manage file systems on storage devices
- D. To swap processes in and out of main memory

Answer: D

Q748. In Android, what is a Binder and what role does it play?

- A. A kernel memory allocation system
- B. An IPC framework using shared memory
- C. A CPU scheduler for Android apps
- D. A file management utility tool

Answer: B

Q749. What is the copy-on-write (COW) optimization used with fork()?

- A. Creating a complete full copy of the parent immediately
- B. Parent and child share pages; copy only on modification
- C. Writing all process data to disk before forking begins
- D. Preventing the child process from writing to any memory

Answer: B

Q750. What is the purpose of the vfork() system call?

- A. Child shares parent space; parent suspends until exec
- B. It creates a virtual process with no memory mapping
- C. It is identical to the standard fork() call
- D. It creates a process with higher priority than parent

Answer: A

Q751. In the Completely Fair Scheduler (CFS) used by Linux, how is CPU time allocated?

- A. Based only on arrival time of each process
- B. Based on vruntime, picking the smallest vruntime
- C. Using fixed time quanta for all processes equally
- D. Using strict static priority scheduling for all

Answer: B

Q752. What is the problem with using SJF in practice?

- A. Next burst length is unknown and must be estimated
- B. It always causes deadlock among the running tasks
- C. It is far too slow for practical use
- D. It does not support multitasking at all in practice

Answer: A

Q753. In exponential averaging for SJF burst prediction, the formula is $\tau(n) = \alpha t(n) + (1-\alpha)\tau(n-1)$. What happens when $\alpha = 0$?

- A. Only the most recent actual burst is considered here
- B. All previous burst lengths are weighted equally here
- C. Recent history has no effect; uses initial estimate only
- D. The prediction always becomes zero in all situations

Answer: C

Q754. In multiprocessor scheduling, what is processor affinity?

- A. Keeping a process on the same processor for cache
- B. The clock speed of a specific processor core
- C. The power consumption of a specific processor
- D. The total number of processors in the system

Answer: A

Q755. What is load balancing in multiprocessor scheduling?

- A. Dividing a single process into equal-sized parts
- B. Running all processes on a single CPU processor
- C. Reducing the CPU clock speed to save energy
- D. Distributing workload evenly across all processors

Answer: D

Q756. What is the difference between asymmetric and symmetric multiprocessing scheduling?

- A. Asymmetric uses more processors than symmetric
- B. They are identical in behavior and design
- C. Symmetric is always faster than asymmetric mode
- D. Asymmetric has one master; symmetric self-schedules

Answer: D

Q757. In real-time CPU scheduling, what is the difference between rate-monotonic and earliest-deadline-first (EDF) scheduling?

- A. Rate-monotonic is non-preemptive; EDF is preemptive
- B. Rate-monotonic uses fixed priority; EDF uses deadlines
- C. EDF uses fixed priority; rate-monotonic uses dynamic
- D. They are identical algorithms for real-time systems

Answer: B

Q758. What is the maximum CPU utilization bound for rate-monotonic scheduling with n processes?

- A. Always exactly 75% utilization
- B. Always 100% utilization
- C. Always exactly 50% utilization
- D. $n(2^{1/n}-1)$, approaches ~69.3%

Answer: D

Q759. What is NUMA-aware scheduling?

- A. Scheduling that ignores all memory placement
- B. Scheduling based on network usage patterns
- C. A scheduling algorithm for non-uniform CPUs
- D. Placing processes near their memory for speed

Answer: D

Q760. What is proportional share scheduling?

- A. Strict priority-based scheduling approach
- B. Scheduling based on memory usage amounts
- C. CPU time proportional to allocated shares
- D. Equal CPU time for all running processes

Answer: C

Q761. What is the many-to-many threading model and what advantage does it offer?

- A. It creates many processes for running threads
- B. It is the simplest threading model to implement
- C. It multiplexes user threads onto kernel threads
- D. It maps many threads to many individual CPUs

Answer: C

Q762. What is the two-level threading model?

- A. A model with exactly two priority levels only
- B. A model with exactly two scheduling queues
- C. Many-to-many plus binding user thread to kernel
- D. A model that uses exactly two processor CPUs

Answer: C

Q763. What is a Lightweight Process (LWP) in the context of user and kernel thread mapping?

- A. A process that runs and completes very quickly
- B. A type of user-level thread implementation
- C. An intermediate structure between user and kernel
- D. A process with very low memory usage footprint

Answer: C

Q764. How does signal handling work in a multithreaded program?

- A. Signals are ignored in multithreaded programs
- B. Only the main thread ever receives any signal
- C. Signals are always delivered to every thread
- D. Signals go to applicable, all, or designated thread

Answer: D

Q765. What is the scheduler activation mechanism?

- A. A method to start or initialize the operating system
- B. A timer-based scheduling technique for real-time use
- C. A way to activate the CPU scheduling component
- D. Kernel upcalls maintaining correct kernel thread count

Answer: D

Q766. What is Grand Central Dispatch (GCD)?

- A. A database system for concurrency
- B. Apple technology managing dispatch queues
- C. A network routing protocol for devices
- D. A Linux kernel module for scheduling

Answer: B

Q767. What is the fork-join parallelism pattern?

- A. A pattern for optimizing disk I/O performance
- B. A pattern for managing file system operations
- C. Parent forks child threads, then joins on completion
- D. A pattern for managing network communication tasks

Answer: C

Q768. What problem does the clone() system call solve in Linux that fork() doesn't?

- A. It creates child processes much faster than fork
- B. It allows more than 256 concurrent running processes
- C. It prevents zombie processes from accumulating in table
- D. It controls exactly which resources parent and child share

Answer: D

Q769. In the context of Intel Threading Building Blocks (TBB), what is a task?

- A. A fixed-size dedicated thread object
- B. A system call for thread management
- C. A work unit mapped to threads by runtime
- D. A hardware timer for task scheduling

Answer: C

Q770. Why have most modern operating systems moved away from the many-to-many threading model to the one-to-one model?

- A. Many-to-many has known security vulnerabilities
- B. Systems now efficiently support many kernel threads
- C. Many-to-many is more expensive to run overall
- D. One-to-one uses significantly less memory overall

Answer: B

Q771. What is the dining philosophers problem and why is it significant?

- A. A problem about memory allocation fragmentation
- B. A problem about restaurant management logistics
- C. A concurrency problem illustrating deadlock and starvation
- D. A problem about network packet routing efficiency

Answer: C

Q772. How can the dining philosophers problem be solved to prevent deadlock?

- A. Add more chopsticks than the total number of diners
- B. Give each philosopher two dedicated chopsticks to use
- C. Limit sitting philosophers, use asymmetric or monitors
- D. Remove one philosopher from the dining table entirely

Answer: C

Q773. What is a read-write lock and when is it preferred over a mutex?

- A. A lock used only for writing to shared data
- B. A lock only used for file read operations
- C. A lock that prevents all access to a resource
- D. Allows concurrent reads but requires exclusive writes

Answer: D

Q774. What is a memory barrier (memory fence)?

- A. A physical barrier inside memory chip hardware
- B. A limit on the total memory allocation size
- C. An instruction enforcing memory operation ordering
- D. A type of virtual memory address translation

Answer: C

Q775. What are lock-free algorithms and why are they important?

- A. Algorithms that use no synchronization at all
- B. Algorithms that run without any operating system
- C. Algorithms using atomic ops like CAS, no locks
- D. Algorithms that are never blocked by anything

Answer: C

Q776. In the readers-writers problem, what is the difference between the first and second variants?

- A. There is no difference between the two at all
- B. First is for single CPU; second for multiprocessor
- C. First uses semaphores; the second uses only monitors
- D. First favors readers no wait; second favors writers

Answer: D

Q777. What is the difference between Hoare-style and Mesa-style monitors?

- A. Hoare signal transfers control; Mesa is just a hint
- B. They use different programming languages entirely
- C. Mesa-style monitors are older than Hoare-style
- D. Hoare-style monitors are faster than Mesa-style

Answer: A

Q778. What is transactional memory?

- A. Memory transferred between running processes
- B. Atomic memory operations similar to databases
- C. A type of CPU cache for fast computations
- D. Memory used for financial transactions only

Answer: B

Q779. What is the ABA problem in lock-free programming?

- A. A deadlock scenario involving two blocked threads
- B. A naming convention issue in the source code
- C. A problem with kernel memory allocation routines
- D. CAS succeeds incorrectly due to A-to-B-to-A change

Answer: D

Q780. What is a futex (fast userspace mutex) in Linux?

- A. User-space mutex with kernel support on contention
- B. A fast file transfer utility for networks
- C. A type of file lock for exclusive access
- D. A network protocol for data synchronization

Answer: A

Q781. In the Banker's Algorithm safety check, what is the time complexity for n processes and m resource types?

- A. $O(n)$ linear
- B. $O(n!)$ factorial
- C. $O(n * m)$ time
- D. $O(n^2 * m)$ time

Answer: D

Q782. What is the problem with resource preemption as a deadlock recovery method?

- A. It uses too much memory for tracking state
- B. Selecting victim, rollback, preventing starvation
- C. It only works on single-processor systems
- D. It runs too fast and is hard to control

Answer: B

Q783. In a resource allocation graph with multiple instances per resource type, a cycle:

- A. Necessary but not sufficient
- B. Never indicates any deadlock
- C. Always indicates deadlock
- D. Neither necessary nor sufficient

Answer: A

Q784. What is a livelock and how does it differ from deadlock?

- A. There is no difference between livelock and deadlock
- B. Livelock only occurs in single-threaded programs only
- C. In livelock processes change state without any progress
- D. Livelock is simply a faster form of deadlock occurring

Answer: C

Q785. How does the 'no preemption' condition get addressed in deadlock prevention for resources like CPU registers and memory?

- A. All held resources are preempted if request fails
- B. The process is immediately terminated by the OS
- C. These resources can never be preempted at all
- D. The resources are duplicated for both processes

Answer: A

Q786. What is the main limitation of the Banker's Algorithm in practice?

- A. It runs too fast for real world use cases
- B. It only works with exactly two processes
- C. Processes rarely know max needs in advance
- D. It does not detect any resource cycles found

Answer: C

Q787. How frequently should a deadlock detection algorithm be invoked?

- A. Only when the system is shutting down
- B. Depends on frequency and tradeoff of cost
- C. Every single millisecond without exception
- D. Once at initial system startup only ever

Answer: B

Q788. In distributed systems, what additional challenges does deadlock detection face?

- A. Deadlocks cannot occur in distributed systems
- B. Only overall performance is affected here
- C. No additional challenges at all exist
- D. Delays, no global state, phantom deadlocks

Answer: D

Q789. What is a two-phase locking protocol and how does it relate to deadlocks?

- A. A protocol using only two shared resources
- B. A protocol with exactly two locks only
- C. Growing acquires, shrinking releases; can deadlock
- D. A protocol that is completely deadlock-free

Answer: C

Q790. What is the wound-wait scheme for deadlock prevention in database systems?

- A. Older wounds younger; younger waits for older
- B. A recovery technique for crashed transactions
- C. A method to wound and terminate processes
- D. A scheduling algorithm for query processing

Answer: A

Q791. What is the working set model and how does it relate to thrashing?

- A. It is a disk scheduling model for I/O requests
- B. It is a CPU scheduling model for processes
- C. It models CPU utilization across all processors
- D. Defines active pages; fewer frames causes thrashing

Answer: D

Q792. What is a multi-level page table and why is it used?

- A. A page table with multiple entries per page
- B. Hierarchical structure saving memory for sparse spaces
- C. A page table stored entirely on secondary disk
- D. A table shared by multiple running processes

Answer: B

Q793. What is an inverted page table?

- A. A page table stored in reverse order
- B. One entry per physical frame, reducing overhead
- C. A page table stored entirely on the disk drive
- D. A table that maps physical to logical addresses

Answer: B

Q794. What is the second-chance (clock) page replacement algorithm?

- A. Modified FIFO checking reference bit, gives second chance
- B. An algorithm that always replaces the second loaded page
- C. A two-pass algorithm scanning memory blocks twice each
- D. A modified version of the FIFO algorithm only

Answer: A

Q795. What is copy-on-write in the context of virtual memory?

- A. Copying all pages to disk during every write
- B. Writing data twice to provide data redundancy
- C. Sharing pages until a write, then making a copy
- D. Copying all pages before any read operation

Answer: C

Q796. What are huge pages (large pages) and when are they beneficial?

- A. Pages that span across multiple physical disk drives
- B. Pages assigned the highest priority access level
- C. Pages used to store very large files on disk
- D. Larger pages reducing TLB misses for big workloads

Answer: D

Q797. What is memory-mapped I/O?

- A. Mapping device registers into address space
- B. A technique for allocating memory to tasks
- C. An I/O scheduling algorithm for requests
- D. Storing I/O data in regular memory buffers

Answer: A

Q798. What is the page-fault frequency (PFF) approach to preventing thrashing?

- A. Setting a maximum number of page faults allowed
- B. Monitoring fault rate and adjusting frame allocation
- C. Counting total page faults across all processes
- D. Preventing all page faults from occurring ever

Answer: B

Q799. What is the buddy system for memory allocation?

- A. A system with two separate pools for allocation
- B. A system where two processes share memory pool
- C. A system for backing up memory to the disk
- D. Power-of-2 blocks, splitting and coalescing buddies

Answer: D

Q800. What is the slab allocator used in Linux kernel memory management?

- A. A page replacement algorithm for virtual memory
- B. A disk allocation method for file storage
- C. A virtual memory technique for address mapping
- D. Pre-allocating object caches to reduce overhead

Answer: D

Q801. What is a journaling file system and how does it improve reliability?

- A. Logs changes to journal before applying for crash recovery
- B. A read-only file system that never allows modifications
- C. A file system that keeps a diary of user activities
- D. A file system that journals all network activity on disk

Answer: A

Q802. What is a log-structured file system (LFS)?

- A. Writes all data sequentially to a log for write speed
- B. A file system that only stores system log files
- C. A file system with built-in logging capability
- D. A file system used only for system log storage

Answer: A

Q803. In UNIX, what is the purpose of the superblock?

- A. To store the largest file in the system
- B. To store file system metadata like size and counts
- C. To speed up file access through data caching
- D. To store user passwords and authentication info

Answer: B

Q804. How do UNIX file systems handle files larger than what direct block pointers in an inode can address?

- A. By splitting the file into multiple smaller separate files
- B. Using indirect, double indirect, and triple indirect pointers
- C. They cannot handle files larger than direct pointers allow
- D. By compressing the file data to reduce total block count

Answer: B

Q805. What is a copy-on-write file system like ZFS or Btrfs?

- A. Never overwrites data; writes to new locations always
- B. A file system that prevents all write operations
- C. A file system that duplicates all data for safety
- D. A file system that copies files when writing to disk

Answer: A

Q806. What is the Network File System (NFS)?

- A. A database for networked storage
- B. A firewall for file server protection
- C. A local file system for one machine
- D. A protocol for remote files as local

Answer: D

Q807. What are extents in modern file systems like ext4?

- A. Extra storage space added to disk
- B. External file links to other systems
- C. Contiguous block range by start and length
- D. Extended file attributes for metadata

Answer: C

Q808. What is file system defragmentation and why is it needed?

- A. Checking file system integrity and repairing errors
- B. Breaking files into smaller fragments for storage
- C. Deleting all fragmented files from the disk drive
- D. Reorganizing blocks to be contiguous for performance

Answer: D

Q809. What is the buffer cache in file system implementation?

- A. A buffer used for printer output spooling
- B. Memory caching disk blocks to reduce I/O ops
- C. A buffer used for storing network packet data
- D. A cache for graphical user interface buffers

Answer: B

Q810. What is a RAID system and what is its purpose?

- A. A debugging tool for disk diagnostics
- B. A memory management technique for allocation
- C. Multiple disks combined for performance or reliability
- D. A type of file system for special purposes

Answer: C

Q811. What is wear leveling in SSDs and why is it necessary?

- A. Equalizing read speeds across memory blocks
- B. Distributing physical wear on the drive casing
- C. Distributing write/erase cycles evenly on cells
- D. Leveling the drive in its mounting bracket

Answer: C

Q812. What is the TRIM command in the context of SSDs?

- A. A command to reduce overall file size used
- B. A command to trim long file names shorter
- C. Tells SSD which blocks are free for erasure
- D. A command to resize existing disk partitions

Answer: C

Q813. What is the write amplification problem in SSDs?

- A. Data being written too quickly to the drive
- B. Amplified noise during write operations on the drive
- C. Physical writes exceed logical due to garbage collection
- D. Writing same data multiple times for data redundancy

Answer: C

Q814. What is the difference between RAID 5 and RAID 6?

- A. RAID 6 uses mirroring instead of distributed parity
- B. There is no significant difference between them at all
- C. RAID 6 is faster in overall performance than RAID 5
- D. RAID 6 uses double parity, surviving two disk failures

Answer: D

Q815. What is the I/O scheduler's role in Linux and how do different schedulers (CFQ, deadline, noop) differ?

- A. They all work in an identical way overall
- B. CFQ ensures fairness; deadline prevents starvation; noop for SSDs
- C. They only differ in their names not in actual behavior
- D. They are all deprecated and no longer used in modern Linux

Answer: B

Q816. What is disk striping and how does it improve performance?

- A. Painting visual stripes on the disk surface
- B. Removing unused sectors from the disk drive
- C. Compressing data stored on the disk drive
- D. Distributing data across disks for parallel I/O

Answer: D

Q817. What is the purpose of a disk controller's command queue (NCQ/TCQ)?

- A. To queue commands issued by the user at terminal
- B. To encrypt all disk commands for data security
- C. To reorder pending I/O for reduced head movement
- D. To limit the number of concurrent disk requests

Answer: C

Q818. What is a hot spare disk in RAID configurations?

- A. A disk that runs at higher operating temperature
- B. An unused disk that auto-replaces a failed disk
- C. A backup disk stored off-site for recovery use
- D. A disk storing the most frequently accessed files

Answer: B

Q819. What is the difference between NVMe and SATA interfaces for SSDs?

- A. SATA is faster than NVMe in all standard benchmarks
- B. NVMe uses PCIe for higher bandwidth and lower latency
- C. NVMe only works with traditional hard disk drive units
- D. They are the exact same interface standard overall

Answer: B

Q820. What is the significance of the 4K sector size (Advanced Format) in modern disk drives?

- A. It slows down the disk read and write speed
- B. It reduces overall disk capacity significantly
- C. It is designed only for SSD drives and not HDDs
- D. Larger sectors improve error correction and efficiency

Answer: D

Q821. What is vectored I/O (scatter-gather I/O)?

- A. I/O operations on vector graphics rendering
- B. Broadcasting I/O to multiple devices at once
- C. Single I/O on multiple non-contiguous buffers
- D. Parallel I/O on vector processors for tasks

Answer: C

Q822. What is the purpose of I/O scheduling at the OS level?

- A. To encrypt I/O data for security and privacy
- B. To slow down all I/O operations for safety
- C. To prevent all I/O operations from executing
- D. To order and optimize I/O for throughput and fairness

Answer: D

Q823. What is the kernel I/O subsystem's role in device reservation?

- A. Updating device firmware and driver software
- B. Reserving devices for future hardware purchase
- C. Backing up all device data to secondary storage
- D. Granting exclusive device access to prevent conflicts

Answer: D

Q824. What is the life cycle of an I/O request from a user application to hardware?

- A. Application goes through network stack then to hardware
- B. Application directly accesses hardware with no intermediary
- C. App calls kernel, driver, controller, then hardware device
- D. Application goes through file system then to the hardware

Answer: C

Q825. What is asynchronous I/O and how does it differ from non-blocking I/O?

- A. Non-blocking uses interrupts while async does not
- B. They are the same thing in every single way
- C. Asynchronous I/O is actually a blocking I/O form
- D. Non-blocking returns available; async notifies on completion

Answer: D

Q826. What is IOCTL (I/O Control) used for?

- A. A syscall for device-specific operations beyond read/write
- B. Input and output data formatting utility
- C. I/O error correction and data recovery mechanism
- D. I/O traffic control and bandwidth management tool

Answer: A

Q827. What is zero-copy I/O and why is it important for performance?

- A. I/O that produces no output at all ever
- B. Eliminating unnecessary copies between kernel and user
- C. I/O that uses no disk space for data storage
- D. I/O with zero errors during the transfer process

Answer: B

Q828. What is the role of the select()/poll()/epoll() system calls?

- A. To monitor multiple file descriptors for I/O readiness
- B. To select which CPU scheduling algorithm to apply
- C. To select files to delete from the disk drive
- D. To poll system users for their preference settings

Answer: A

Q829. What is the difference between port-mapped I/O and memory-mapped I/O?

- A. Port-mapped I/O is always faster than memory-mapped I/O
- B. Port-mapped uses separate space; memory-mapped uses memory
- C. Memory-mapped requires special hardware not in modern CPUs
- D. They are identical in every respect and behavior

Answer: B

Q830. What is the transformer model in I/O subsystem layering?

- A. A machine learning model for I/O prediction
- B. A power supply model for device power management
- C. A database model for storing I/O request metadata
- D. Layered approach where each layer transforms requests

Answer: D

Q831. What is Address Space Layout Randomization (ASLR)?

- A. Randomizing data area positions to prevent exploits
- B. A network addressing scheme for distributed nodes
- C. A disk layout optimization for faster read access
- D. A memory allocation algorithm used by OS kernel

Answer: A

Q832. What is a rootkit?

- A. A set of root user access tools for admin
- B. A root user configuration file set used
- C. A system administration utility toolkit
- D. Malware providing hidden privileged access

Answer: D

Q833. What is the concept of security rings (protection rings) in OS architecture?

- A. Hierarchical privilege levels with inner having more
- B. Physical rings around the processor die chip
- C. A type of ring network topology for communication
- D. A form of data encryption using circular key bits

Answer: A

Q834. What is a side-channel attack?

- A. Exploiting physical info like timing or cache behavior
- B. An attack through a secondary network connection
- C. An attack on backup systems stored at off-site
- D. An attack on auxiliary devices connected to system

Answer: A

Q835. What are Spectre and Meltdown vulnerabilities?

- A. Software bugs found in major web browser apps
- B. Hardware flaws exploiting speculative execution to leak data
- C. Network protocol flaws in the TCP/IP stack implementation
- D. File system vulnerabilities found in modern operating systems

Answer: B

Q836. What is mandatory access control (MAC) and how does it differ from discretionary access control (DAC)?

- A. DAC is mandatory and MAC is discretionary
- B. They are the same access control model used
- C. MAC is system-enforced; DAC is owner-controlled
- D. MAC is weaker than DAC in all situations

Answer: C

Q837. What is Role-Based Access Control (RBAC)?

- A. Access control based on user physical location
- B. A type of encryption for access control data
- C. Controlling access based on time of day only
- D. Permissions assigned to roles; users gain via roles

Answer: D

Q838. What is a sandbox in operating system security?

- A. A debugging tool for code analysis testing
- B. A test database for development and staging
- C. Isolated environment restricting program access
- D. A type of malware infection on the system

Answer: C

Q839. What is the Bell-LaPadula security model?

- A. A network security protocol for data transfer
- B. A database security model for SQL prevention
- C. Confidentiality model: no read up, no write down
- D. A physical security model for server rooms

Answer: C

Q840. What is the Biba integrity model and how does it differ from Bell-LaPadula?

- A. Biba focuses on integrity: no read down, no write up
- B. They are the exact same model under different names
- C. Biba is a networking model for packet integrity checks
- D. Biba replaces Bell-LaPadula in all modern OS systems

Answer: A

Q841. In a hybrid kernel architecture, which services typically remain in kernel space?

- A. Only user applications remain inside the kernel space
- B. Networking stacks exclusively run in the kernel space
- C. All device drivers and file systems run in user space
- D. Core IPC and some drivers stay in kernel for speed

Answer: D

Q842. What security vulnerability does the Meltdown attack exploit in CPUs?

- A. It exploits buffer overflow in user application code
- B. It exploits speculative execution to read kernel data
- C. It exploits faulty device drivers in the OS subsystem
- D. It exploits weak encryption in network protocol sets

Answer: B

Q843. What is the key design trade-off in exokernel operating systems?

- A. Exokernels sacrifice all security for maximum speed
- B. They trade app portability for resource efficiency now
- C. Exokernels trade memory capacity for CPU performance
- D. They sacrifice networking for faster disk I/O throughput

Answer: B

Q844. How does a Type-1 hypervisor differ from a Type-2 hypervisor?

- A. Type-2 runs on bare metal while Type-1 needs a host
- B. Type-1 runs directly on bare metal without host OS
- C. Type-1 only supports Linux; Type-2 supports all OSes
- D. Both types require a host operating system to function

Answer: B

Q845. What problem does the UEFI boot process solve over legacy BIOS?

- A. UEFI only supports smaller disk sizes than BIOS can
- B. UEFI is slower but more compatible with older hardware
- C. UEFI eliminates the need for any operating system now
- D. UEFI removes the 2TB disk limit and adds secure boot

Answer: D

Q846. In capability-based OS design, what does a capability represent?

- A. A type of scheduling algorithm for real-time tasks
- B. A measure of CPU processing speed in megahertz
- C. A token granting specific access rights to objects
- D. A hardware register for storing interrupt addresses

Answer: C

Q847. What challenge does the OS face in managing NUMA architectures?

- A. Memory access times vary by processor-memory distance
- B. All memory accesses take the same time in NUMA setups
- C. NUMA systems cannot support more than one processor
- D. NUMA eliminates the need for cache memory in all CPUs

Answer: A

Q848. What is the fundamental principle behind separating mechanism and policy?

- A. It ensures the OS can never be modified or updated
- B. It prevents any user from customizing the OS at all
- C. It allows changing decisions without changing the code
- D. It requires all policies to be hard-coded in hardware

Answer: C

Q849. What is a unikernel and how does it differ from a traditional OS?

- A. A unikernel bundles app and OS in one address space
- B. A unikernel is a multi-user general-purpose OS now
- C. A unikernel requires a full OS installation underneath
- D. A unikernel runs many applications simultaneously here

Answer: A

Q850. Why is the halting problem relevant to operating system design?

- A. It guarantees all programs terminate in finite time span
- B. It proves the OS can always detect all infinite loops
- C. It shows no algorithm can decide if programs will halt
- D. It applies only to database management systems today

Answer: C

Q851. In the L4 microkernel family, what design decision improves IPC performance?

- A. L4 uses asynchronous message passing exclusively always
- B. L4 batches all IPC messages and processes them in bulk
- C. L4 implements synchronous IPC with register-based transfer
- D. L4 eliminates IPC entirely and uses shared memory only

Answer: C

Q852. What is the key challenge in defining layers for a layered OS?

- A. Finding hardware that supports the layered architecture
- B. Determining correct layer ordering with dependencies here
- C. Making sure each layer has the exact same code size now
- D. Ensuring every layer can communicate with all other ones

Answer: B

Q853. How does the Mach microkernel handle memory management?

- A. It requires only hardware-specific memory management
- B. It manages all memory entirely within the microkernel
- C. It does not support virtual memory in any form at all
- D. It uses external pagers in user space for page handling

Answer: D

Q854. What is a system generation (SYSGEN) process in OS deployment?

- A. It automatically writes new operating system source code
- B. It generates documentation for the operating system APIs
- C. It configures the OS for specific hardware at installation
- D. It creates backup copies of the OS kernel on other disk

Answer: C

Q855. Why do modern OS designs favor hybrid kernel over pure microkernel?

- A. Pure microkernels provide better performance in all cases
- B. Hybrid kernels balance modularity with better performance
- C. Pure microkernels cannot support any device drivers now
- D. Hybrid kernels are simpler to implement than microkernels

Answer: B

Q856. What is the role of the Hardware Abstraction Layer in Windows NT?

- A. HAL manages user interface themes and visual appearance
- B. HAL compiles C++ code into native machine instructions
- C. HAL hides hardware differences from upper OS layers now
- D. HAL provides a game rendering engine for DirectX apps

Answer: C

Q857. How does the Android architecture differ from standard Linux?

- A. Android adds middleware like ART runtime over Linux kernel
- B. Android does not use any Linux components in its design
- C. Android replaces the Linux kernel with a custom one here
- D. Android uses Linux only for networking and nothing else

Answer: A

Q858. What problem does eBPF solve in modern Linux kernels?

- A. It replaces the TCP/IP stack with a new protocol stack
- B. It allows safe programmable extensions in the kernel now
- C. It provides a graphical desktop environment for Linux
- D. It manages Bluetooth device connections in Linux only

Answer: B

Q859. What is the significance of MINIX 3 in operating system research?

- A. MINIX 3 eliminated the need for file systems completely
- B. MINIX 3 proved monolithic kernels are always superior
- C. MINIX 3 demonstrated a reliable self-healing microkernel
- D. MINIX 3 was the first OS to support graphical interfaces

Answer: C

Q860. How do C library wrappers simplify system call invocation?

- A. They bypass the kernel entirely for all OS operations
- B. They provide function interface hiding assembly details
- C. They convert system calls into network protocol packets
- D. They eliminate the need for privilege mode transitions

Answer: B

Q861. How does copy-on-write optimize the fork() system call?

- A. It eliminates the need for virtual memory addressing
- B. It copies the entire address space at fork instantly
- C. It prevents child processes from writing any memory
- D. It shares pages until a write then copies that page

Answer: D

Q862. What is vfork() and how does it differ from fork()?

- A. vfork() creates multiple child processes at same time
- B. vfork() bypasses the kernel for process creation now
- C. vfork() is identical to fork() in all design aspects
- D. vfork() shares address space and suspends the parent

Answer: D

Q863. How do process groups and sessions work in Unix systems?

- A. Sessions and process groups are the same exact concept
- B. Every process belongs to exactly one group and session
- C. Process groups exist only for root user processes now
- D. Process groups are only used in Windows and not Unix

Answer: B

Q864. How do preemptive and cooperative multitasking differ?

- A. Preemptive lets OS forcibly take CPU from a process
- B. Cooperative lets the OS forcibly interrupt any process
- C. Both approaches give unlimited CPU time to processes
- D. Preemptive requires processes to yield voluntarily now

Answer: A

Q865. What is the purpose of the clone() system call in Linux?

- A. It converts a user process into a kernel-level process
- B. It merges two existing processes into a single process
- C. It creates a thread or process with fine-grained sharing
- D. It only duplicates files on disk, not any processes

Answer: C

Q866. How does Chrome use process architecture for security?

- A. Chrome uses separate processes per tab for isolation
- B. Chrome runs all tabs in one single process for speed
- C. Chrome does not use processes; it uses threads only
- D. Chrome relies only on the OS kernel for tab isolation

Answer: A

Q867. What is the role of the init process (PID 1) in Unix?

- A. It only manages network interfaces during boot time
- B. It handles only file system mounting and nothing else
- C. It is temporary and exits after boot process completes
- D. It is ancestor of all processes and adopts all orphans

Answer: D

Q868. What challenges arise with process migration in distributed systems?

- A. Only the process ID needs to move between machines now
- B. Process migration is trivial with no challenges at all
- C. Open file handles and network state must be transferred
- D. Process migration eliminates the need for scheduling here

Answer: C

Q869. How does process creation differ between Windows and Unix?

- A. Windows maintains a strict parent-child tree like Unix
- B. Neither system tracks relationships between processes now
- C. Unix has parent-child trees; Windows uses handle tokens
- D. Both systems have identical process creation mechanisms

Answer: C

Q870. What is process checkpointing in fault-tolerant systems?

- A. It saves process state periodically for failure recovery
- B. It permanently deletes process state to free memory
- C. It checks if a process has any source code syntax errors
- D. It counts the number of processes running on the system

Answer: A

Q871. How does the Completely Fair Scheduler in Linux work?

- A. CFS maintains virtual runtime for fair CPU allocation
- B. CFS uses lottery scheduling for random CPU assignment
- C. CFS uses strict priority queues with fixed time slices
- D. CFS implements simple Round Robin with a fixed quantum

Answer: A

Q872. What is the exponential averaging formula for burst prediction?

- A. Next burst is always set to a fixed constant value here
- B. Next burst equals the exact length of the previous burst
- C. $\tau(n+1) = \alpha \cdot t(n) + (1-\alpha) \cdot \tau(n)$ using history
- D. Prediction uses only the average of all prior burst times

Answer: C

Q873. What is priority inversion and how can it be solved?

- A. Low-priority blocks high-priority; use priority inheritance
- B. High-priority tasks always run first without any exception
- C. Priority inversion occurs only in single-core CPU systems
- D. It is solved by removing all priorities from the system

Answer: A

Q874. How does the O(1) scheduler differ from CFS in Linux?

- A. O(1) is fairer than CFS for interactive desktop workloads
- B. O(1) uses active/expired arrays; CFS uses red-black tree
- C. Both schedulers are identical in design and implementation
- D. CFS uses fixed time slices while O(1) uses virtual runtime

Answer: B

Q875. What is gang scheduling in multiprocessor systems?

- A. It only works with single-threaded processes on one core
- B. It co-schedules related threads simultaneously on CPUs
- C. It randomly assigns processes to processors without order
- D. It schedules one process at a time across all CPUs now

Answer: B

Q876. How does processor affinity affect multiprocessor scheduling?

- A. It has no effect on scheduling in multiprocessor systems
- B. It forces processes to migrate between CPUs constantly
- C. It keeps a process on the same CPU to exploit warm cache
- D. It assigns all processes to the same single CPU core here

Answer: C

Q877. What is global vs. partitioned scheduling in multicore systems?

- A. Global and partitioned scheduling are the exact same thing
- B. Global scheduling assigns each process to a fixed core
- C. Partitioned scheduling uses a single queue for all cores
- D. Global uses one shared queue; partitioned uses per-core

Answer: D

Q878. What is rate-monotonic scheduling for real-time systems?

- A. It uses Round Robin for all periodic real-time task sets
- B. It assigns the same fixed priority to every task equally
- C. It assigns priorities randomly to all real-time task sets
- D. It assigns priorities inversely proportional to periods

Answer: D

Q879. How does Earliest Deadline First schedule real-time tasks?

- A. It only works with non-periodic tasks in batch systems
- B. It dynamically prioritizes tasks with nearest deadlines
- C. It schedules the task with the latest deadline first
- D. It ignores deadlines and uses burst time for scheduling

Answer: B

Q880. What is the schedulability test for rate-monotonic scheduling?

- A. All tasks must have priority higher than the OS kernel
- B. Each task must have the same exact period and burst time
- C. CPU utilization must be below 100 percent for all tasks
- D. Sum of C_i/T_i must be at most $n \cdot (2^{1/n} - 1)$ for guarantee

Answer: D

Q881. What is a lightweight process (LWP) in many-to-many threading?

- A. An LWP is a file storing thread state on the hard disk
- B. An LWP is intermediate between user and kernel threads
- C. An LWP is a network packet for inter-thread messaging
- D. An LWP is a physical CPU core in the processor chip

Answer: B

Q882. What is the scheduler activation mechanism in threading?

- A. Kernel notifies user library via upcalls about events
- B. It is a simple timer interrupt for thread scheduling
- C. It replaces the CPU scheduler with a user-space one
- D. It prevents communication between kernel and threads

Answer: A

Q883. How does Windows thread scheduling differ from Linux?

- A. Both systems only schedule processes, never any threads
- B. Windows schedules processes; Linux schedules threads
- C. Windows uses Round Robin; Linux uses SJF exclusively
- D. Windows schedules threads; Linux schedules tasks via CFS

Answer: D

Q884. What is the thundering herd problem in threaded servers?

- A. A single thread monopolizes the CPU preventing others
- B. Threads run too slowly due to excessive synchronization
- C. Too many threads cause total memory exhaustion now
- D. Multiple threads wake for one event but one handles it

Answer: D

Q885. What is the purpose of the futex mechanism in Linux?

- A. It provides a graphics rendering pipeline for threads
- B. It controls CPU frequency scaling for threaded workload
- C. It manages file transfers between threads over network
- D. It implements fast user-space locking with kernel fallback

Answer: D

Q886. How does hyperthreading affect OS thread scheduling?

- A. It eliminates the need for any software threading at all
- B. It reduces the number of CPUs visible to the OS kernel
- C. It has no effect on how the OS schedules threads at all
- D. OS sees additional logical CPUs and schedules on them now

Answer: D

Q887. What are the challenges of implementing user-level threads?

- A. User-level threads are always slower than kernel threads
- B. User-level threads cannot be created by any application
- C. User-level threads require special hardware support only
- D. A blocking syscall blocks all threads in the process now

Answer: D

Q888. What is the two-level model in thread implementation?

- A. It combines many-to-many with one-to-one mapping
- B. It requires two separate kernels for thread management
- C. It uses exactly two threads per process, no more here
- D. It maps all threads to exactly two kernel-level threads

Answer: A

Q889. How does Grand Central Dispatch handle threading on macOS?

- A. GCD only supports single-threaded application execution
- B. GCD requires programmers to manually create threads
- C. GCD uses blocks and dispatch queues managed by runtime
- D. GCD is a hardware threading mechanism in Apple chips

Answer: C

Q890. What is a fiber and how does it differ from a thread?

- A. Fibers are identical to threads in every way possible
- B. Fibers are hardware threads built into processor cores
- C. Fibers are cooperatively scheduled user-level threads
- D. Fibers are kernel-level threads with preemptive scheduling

Answer: C

Q891. What is the dining philosophers problem and its significance?

- A. It models network routing algorithms in large data centers
- B. It is about optimizing database query execution plans now
- C. It models resource allocation leading to potential deadlock
- D. It describes scheduling of philosophers teaching classes

Answer: C

Q892. How does Read-Copy-Update (RCU) achieve lock-free reads?

- A. RCU prevents any writes from occurring while reading
- B. RCU uses spinlocks internally for all read operations
- C. RCU locks all readers while writer updates the data
- D. Readers access old data while writers prepare new copy

Answer: D

Q893. What is the ABA problem in lock-free data structures?

- A. ABA is an algorithm for balanced binary tree building
- B. It describes automatic backup and archival of data
- C. It refers to alphabetical ordering of lock acquisition
- D. Value changes from A to B to A, hiding the updates

Answer: D

Q894. What is a memory barrier instruction used for?

- A. It ensures memory operations complete in proper order
- B. It prevents the CPU from accessing some memory regions
- C. It doubles available memory by compressing all data now
- D. It creates a physical barrier in RAM to separate data

Answer: A

Q895. What differs between Mesa and Hoare monitor semantics?

- A. Hoare uses signal-and-continue; Mesa signal-and-wait
- B. Mesa requires busy waiting; Hoare uses blocking only
- C. Mesa and Hoare semantics are completely identical here
- D. Mesa uses signal-and-continue; Hoare signal-and-wait

Answer: D

Q896. How does transactional memory simplify concurrent programming?

- A. It only works with single-threaded programs in practice
- B. It requires more locks than traditional synchronization
- C. It eliminates all shared memory between threads fully
- D. It lets code blocks execute atomically without locks now

Answer: D

Q897. What is the priority ceiling protocol and how does it work?

- A. Each mutex gets ceiling equal to highest priority user now
- B. It only lets highest priority thread acquire locks at all
- C. It lowers all thread priorities to zero to stop conflicts
- D. It eliminates all priorities making every thread equal here

Answer: A

Q898. What is lock-free vs. wait-free in concurrent algorithms?

- A. Wait-free ensures system progress but not per-thread
- B. Lock-free ensures system progress; wait-free per-thread
- C. Lock-free and wait-free algorithms are exactly the same
- D. Lock-free uses hidden locks but hides them from users

Answer: B

Q899. What does the sleeping barber problem model?

- A. It models network packet queuing in router memory now
- B. It describes file system journaling for crash recovery
- C. A scheduling algorithm for barber shop appointments
- D. It models bounded-buffer producer-consumer with sleep

Answer: D

Q900. How does Linux handle preemption within critical sections?

- A. Linux disables preemption using preemption counters now
- B. Linux never allows any kernel-level preemption at all
- C. Linux always allows preemption inside critical sections
- D. Linux uses hardware locks to prevent all preemption

Answer: A

Q901. What is the time complexity of the Banker's algorithm?

- A. $O(n*n*m)$ for determining safe sequence each time here
- B. $O(n)$ linear time complexity for each resource request
- C. $O(\log n)$ logarithmic time for each allocation decision
- D. $O(1)$ constant time regardless of process or resources

Answer: A

Q902. How does the Linux kernel handle deadlock in practice?

- A. Linux uses lockdep to detect potential deadlock patterns
- B. Linux ignores deadlocks and restarts the system if stuck
- C. Linux prevents all four deadlock conditions at all times
- D. Linux implements full Banker's algorithm at runtime

Answer: A

Q903. What is two-phase locking's relationship to deadlock?

- A. Two-phase locking completely eliminates all deadlocks
- B. Two-phase locking causes deadlock in every single case
- C. Two-phase locking ensures serializability allows deadlock
- D. Two-phase locking has no relationship to deadlock at all

Answer: C

Q904. What is a knot in a resource allocation graph?

- A. A knot is simply another name for any edge in graph
- B. A knot means the graph has no cycles present at all
- C. A cycle where no outside process can break the cycle
- D. A knot indicates that the system is in a safe state

Answer: C

Q905. How does wound-wait differ from wait-die for databases?

- A. Wait-die: older waits or dies; wound-wait wounds younger
- B. Both schemes are identical and behave the same always
- C. Wound-wait: older wounds younger; wait-die: older waits
- D. Wound-wait preempts younger; wait-die makes older wait

Answer: C

Q906. How are victim selection strategies used in deadlock recovery?

- A. Cost-based selection minimizes rollback overhead impact
- B. Victim selection only applies to file system operations
- C. All victims are always the newest process in the system
- D. Victim selection is random and cannot be optimized now

Answer: A

Q907. How can deadlock occur in distributed systems?

- A. Processes on different nodes can form distributed cycles
- B. Deadlock cannot occur across multiple distributed nodes
- C. Only single-node deadlocks are possible in any system
- D. Distributed systems automatically prevent all deadlocks

Answer: A

Q908. What is the phantom deadlock problem in distributed detection?

- A. It describes deadlocks that only occur during system startup
- B. It is a deadlock that is invisible to all detection methods
- C. It occurs when a detected deadlock does not actually exist
- D. It refers to deadlocks in virtual machine environments only

Answer: C

Q909. What is hierarchical deadlock prevention using lock ordering?

- A. It assigns random order to locks at runtime dynamically
- B. It groups locks into levels and enforces strict ordering
- C. It eliminates all locks from the system to prevent issues
- D. It uses only one global lock for the entire system here

Answer: B

Q910. How does the edge-chasing algorithm detect distributed deadlocks?

- A. It uses a centralized server to collect all wait-for info
- B. It prevents deadlock rather than detecting it after fact
- C. It broadcasts all lock info to every node all the time
- D. It propagates probe messages along wait-for edges across

Answer: D

Q911. How does the inverted page table reduce memory usage?

- A. It eliminates the need for any page table entirely
- B. It uses one entry per virtual page for each process
- C. It compresses page table entries to half their size
- D. It uses one entry per physical frame not per page

Answer: D

Q912. What is the clock (second-chance) page replacement algorithm?

- A. It gives each page a second chance via reference bit
- B. It always replaces the page with highest frame number
- C. It schedules page replacements at fixed clock intervals
- D. It replaces pages based on the actual wall clock time

Answer: A

Q913. What is copy-on-write in virtual memory management?

- A. Copying entire memory space when a process is created
- B. Sharing pages until write occurs then making private copy
- C. Writing all memory contents to disk after every instruction
- D. Preventing any process from writing to memory at all

Answer: B

Q914. How does ASLR improve security?

- A. It disables virtual memory to improve system performance
- B. It encrypts all data stored in process address space
- C. It randomizes code and data locations to prevent exploits
- D. It prevents all processes from accessing shared memory

Answer: C

Q915. What is the optimal page replacement algorithm's limitation?

- A. It does not support virtual memory in any implementation
- B. It causes more page faults than any other algorithm
- C. It can only work with a single page frame in system
- D. It requires future knowledge of page references to work

Answer: D

Q916. How does huge page support improve TLB performance?

- A. Huge pages reduce TLB size to improve lookup speed
- B. Huge pages reduce TLB miss frequency with fewer entries
- C. Huge pages eliminate the need for any TLB hardware now
- D. Huge pages make TLB slower but more accurate in lookup

Answer: B

Q917. What is the page-buffering algorithm used for?

- A. Buffering network data in page-sized blocks only
- B. Compressing pages to fit more data into memory now
- C. Preventing page faults by locking all pages in RAM
- D. Keeping a pool of free frames to speed replacement

Answer: D

Q918. What is memory-mapped I/O and how does it work?

- A. It stores all I/O data exclusively in the CPU registers
- B. It maps physical memory addresses to network addresses
- C. It maps I/O device registers into memory address space
- D. It eliminates the need for any I/O devices in system

Answer: C

Q919. What is kernel same-page merging (KSM) in Linux?

- A. It merges all kernel modules into a single binary
- B. It identifies and merges identical pages across procs
- C. It prevents any page sharing between virtual machines
- D. It splits each page into two smaller pages for speed

Answer: B

Q920. How does the buddy system manage memory allocation?

- A. It assigns memory randomly without any structure here
- B. It pairs every process with a buddy for backup now
- C. It splits memory into power-of-two blocks recursively
- D. It allocates memory in fixed blocks without splitting

Answer: C

Q921. How does ext4 use extents instead of block pointers?

- A. Extents are only used for files smaller than 4 KB
- B. An extent describes a contiguous run with one entry
- C. Extents eliminate the need for any metadata on disk
- D. Extents store each block address individually here

Answer: B

Q922. What is a copy-on-write file system and its advantage?

- A. It copies all data twice for redundancy on every write
- B. It writes data in place and copies only on deletion
- C. It never overwrites existing data ensuring consistency
- D. It only allows reads and prevents all write operations

Answer: C

Q923. How does the B+ tree structure benefit NTFS?

- A. It provides efficient $O(\log n)$ lookups and range scans
- B. It only works with files smaller than one megabyte now
- C. It stores file data directly inside B-tree leaf nodes
- D. It eliminates the need for any disk space management

Answer: A

Q924. What is the log-structured file system design philosophy?

- A. LFS stores data in traditional inode-based tree layout
- B. LFS writes all data and metadata sequentially to log
- C. LFS requires all data to be stored in main memory now
- D. LFS uses random writes to maximize disk head movement

Answer: B

Q925. How does ZFS ensure data integrity using checksums?

- A. ZFS checksums are stored alongside data in same block
- B. ZFS stores checksums in parent blocks enabling healing
- C. ZFS does not use any checksums for integrity checking
- D. ZFS only checksums metadata, not actual file data now

Answer: B

Q926. What is the TRIM command for SSDs in file systems?

- A. It trims the physical size of the SSD device now
- B. It informs SSD which blocks are no longer in use
- C. It compresses data on SSD to save storage space
- D. It increases read speed by defragmenting the SSD

Answer: B

Q927. How do deduplication techniques work in enterprise storage?

- A. They duplicate important files for safety purposes
- B. They compress each file individually to reduce size
- C. They identify and eliminate duplicate blocks to save
- D. They remove all files not accessed in recent months

Answer: C

Q928. What differs between ordered and writeback journaling?

- A. Ordered mode does not use a journal; writeback does
- B. Writeback writes data first; ordered metadata first now
- C. Both modes write data and metadata in the same order
- D. Ordered writes data before metadata; writeback either

Answer: D

Q929. What is the purpose of the superblock in Unix file systems?

- A. It stores the boot loader program for the OS kernel
- B. It contains critical file system metadata like size now
- C. It stores the contents of the largest file in system
- D. It is extra storage reserved for the root user only

Answer: B

Q930. How does FUSE work in Linux?

- A. FUSE moves all file system code into kernel space
- B. FUSE allows implementing file systems in user space
- C. FUSE replaces the VFS layer with a simpler version
- D. FUSE is a hardware device for connecting ext drives

Answer: B

Q931. How does LOOK improve upon the SCAN algorithm?

- A. LOOK eliminates the need for any head movement
- B. LOOK moves head to the physical end of disk
- C. LOOK reverses at the last request, not disk end
- D. LOOK processes requests in strict FCFS order now

Answer: C

Q932. What is the write amplification problem in SSDs?

- A. Writes are amplified to improve SSD performance
- B. SSDs amplify signals to write data faster on chip
- C. Writes are duplicated to multiple chips for safety
- D. More physical data is written than logical requested

Answer: D

Q933. How does RAID 6 improve on RAID 5 fault tolerance?

- A. RAID 6 uses triple mirroring of all data blocks
- B. RAID 6 uses dual parity allowing two disk failures
- C. RAID 6 stores four copies of parity on one drive
- D. RAID 6 and RAID 5 offer exactly the same safety

Answer: B

Q934. What is the FTL in solid-state drives?

- A. A network layer for transferring data between SSDs
- B. A physical layer connecting SSD to motherboard
- C. Firmware mapping logical addresses to flash pages
- D. A software tool for formatting flash drives on PC

Answer: C

Q935. What is shingled magnetic recording for HDDs?

- A. It uses circular tracks with no overlap at all
- B. It uses multiple read heads for faster read speed
- C. Tracks overlap like shingles for higher density now
- D. It eliminates platters using solid-state memory

Answer: C

Q936. What is NCQ in SATA drives?

- A. It prevents commands from executing out of order
- B. It queues user interface commands for the OS now
- C. It lets drives reorder commands for optimal execution
- D. It queues network commands for remote disk access

Answer: C

Q937. How does NVMe improve upon AHCI for SSDs?

- A. NVMe uses the same command set as legacy IDE chips
- B. NVMe and AHCI provide identical performance for SSDs
- C. NVMe uses deeper queues and lower latency than AHCI
- D. NVMe is designed for spinning disks limiting SSDs

Answer: C

Q938. What is the NOOP I/O scheduler and when is it used?

- A. NOOP completely eliminates all I/O scheduling here
- B. NOOP is the most advanced scheduler for disk drives
- C. NOOP uses simple FIFO queue with minimal ordering
- D. NOOP performs extensive sorting of I/O for HDDs

Answer: C

Q939. What is the RAID write hole problem?

- A. Power failure during write leaves parity inconsistent
- B. It prevents any write from completing on RAID now
- C. It is a physical hole in disk platter causing damage
- D. It only occurs when reading data from RAID array now

Answer: A

Q940. How does host-managed zoned storage work with ZNS SSDs?

- A. SSD firmware manages zones without any host input
- B. ZNS SSDs work identically to traditional block SSDs
- C. The host has no control over data placement here
- D. Host manages zones and writes sequentially in them

Answer: D

Q941. How does scatter-gather DMA improve I/O performance?

- A. It transfers data to non-contiguous memory regions now
- B. It eliminates DMA and uses programmed I/O instead
- C. It gathers all data into single buffer before transfer
- D. It scatters data randomly across memory for security

Answer: A

Q942. What is the IOCP model in Windows?

- A. A port number used for network socket communication
- B. A scalable model for handling many async I/O events
- C. A model that blocks all threads until I/O finishes
- D. A physical port for connecting I/O devices to PC

Answer: B

Q943. How does Linux epoll improve over select()?

- A. epoll uses O(1) notification instead of O(n) scanning
- B. epoll only works with network sockets not file I/O
- C. epoll and select have identical performance overall
- D. epoll monitors fewer file descriptors than select

Answer: A

Q944. What is IOMMU and how does it improve I/O security?

- A. It is a type of MMU used exclusively for memory mgmt
- B. It encrypts all data transferred between devices and CPU
- C. It translates device addresses providing memory isolation
- D. It is a memory unit storing I/O driver code only

Answer: C

Q945. How do edge-triggered and level-triggered I/O notifications differ?

- A. Edge-triggered notifications are always slower than level
- B. Both notification types trigger in the same situation here
- C. Level-triggered only works with sockets not with disk I/O
- D. Edge notifies on state transition; level while state holds

Answer: D

Q946. What is the io_uring interface in modern Linux?

- A. It provides high-performance async I/O via ring buffers
- B. It creates circular network connections between nodes
- C. It manages circular disk platters in HDD controllers
- D. It is a graphical interface for configuring I/O drivers

Answer: A

Q947. How does RDMA work?

- A. It requires CPU to copy all data for remote transfers
- B. It allows direct memory access between remote machines
- C. It only works within a single machine for local DMA
- D. It uses traditional TCP/IP for remote memory operations

Answer: B

Q948. What is interrupt coalescing and why is it used?

- A. It eliminates all interrupts using only polling for I/O
- B. It combines multiple events into fewer interrupt signals
- C. Each I/O event generates a separate interrupt always
- D. It splits one interrupt into multiple signals for detail

Answer: B

Q949. What is the SPDK approach to storage?

- A. It uses user-space drivers and polling for low latency
- B. It is a standard kernel-based storage driver framework
- C. It is a hardware kit for assembling storage arrays now
- D. It provides a graphical toolkit for storage management

Answer: A

Q950. How does virtio enable efficient I/O virtualization?

- A. It uses hardware passthrough without any abstraction
- B. It eliminates the need for any I/O drivers in the VM
- C. It provides standard paravirtual I/O interface for VMs
- D. It emulates physical hardware perfectly for each VM

Answer: C

Q951. How does return-oriented programming bypass NX protection?

- A. ROP disables the NX bit in CPU page table entries
- B. ROP chains existing code gadgets to execute payload
- C. ROP injects new executable code into stack directly
- D. ROP only works on systems without memory protection

Answer: B

Q952. What is the Trusted Platform Module and its security role?

- A. A hardware chip providing cryptographic security ops
- B. A kernel module managing user authentication only
- C. A software module scanning for viruses on system
- D. A network module filtering traffic like a firewall

Answer: A

Q953. How does mandatory access control differ from discretionary?

- A. MAC uses system-enforced labels; DAC lets owners control
- B. MAC and DAC are different names for the same mechanism
- C. Both give resource owners full control over all access
- D. MAC lets owners set permissions; DAC uses system policy

Answer: A

Q954. What is a side-channel attack and how does Spectre work?

- A. Spectre exploits speculative execution via cache timing
- B. Spectre exploits buffer overflows in browser apps only
- C. It attacks through unused network ports on system
- D. Side-channel attacks directly modify target source code

Answer: A

Q955. How does SELinux implement mandatory access control?

- A. SELinux assigns security labels to subjects and objects
- B. SELinux only encrypts file system data for privacy
- C. SELinux manages firewalls but not file access at all
- D. SELinux replaces Linux kernel with a secure version

Answer: A

Q956. What is a zero-day vulnerability?

- A. A flaw affecting zero users due to limited deployment
- B. A flaw existing only on the day software is released
- C. A flaw known for zero days and still unpatched now
- D. A flaw taking zero effort to exploit using any tool

Answer: C

Q957. How does control flow integrity protect against code-reuse?

- A. CFI encrypts all control flow instructions in binary
- B. CFI ensures execution follows predetermined control graph
- C. CFI only protects against buffer overflow not ROP now
- D. CFI prevents any branching instructions from executing

Answer: B

Q958. What is the confused deputy problem in security?

- A. A user confused about which password to use for login
- B. A privileged program tricked into misusing its authority
- C. An admin who misconfigures the security settings now
- D. A process confusing virtual and physical memory addrs

Answer: B

Q959. How does memory tagging (MTE) prevent memory safety bugs?

- A. MTE tags memory with metadata detecting out-of-bounds
- B. MTE increases the amount of physical RAM available here
- C. MTE encrypts all memory contents using hardware now
- D. MTE only works with interpreted languages not compiled

Answer: A

Q960. What is attestation in trusted computing?

- A. A technique for testing software under heavy load stress
- B. A legal document certifying software license compliance
- C. Cryptographic proof that system is in a known trusted state
- D. A user declaring identity through a sworn statement now

Answer: C

Q961. What is the key difference between a Type-0 and a Type-2 hypervisor in virtualization?

- A. Type-0 is firmware-based while Type-2 runs as an application on a host OS
- B. Type-0 is software while Type-2 is hardware
- C. Type-0 supports only Linux while Type-2 supports only Windows
- D. Type-0 is slower than Type-2 in all scenarios

Answer: A

Q962. What is paravirtualization and how does it improve VM performance?

- A. Guests are modified to make hypercalls instead of privileged instructions, reducing overhead
- B. Guests run unmodified but the hypervisor interprets all instructions
- C. The host OS is removed and guests run directly on hardware
- D. Multiple VMs share the same guest operating system image

Answer: A

Q963. What role does the IOMMU play in modern computer architectures?

- A. It translates device I/O addresses to physical memory addresses and provides isolation
- B. It manages CPU cache coherence across multiple cores
- C. It schedules interrupt handling among processors
- D. It encrypts all data transferred between CPU and memory

Answer: A

Q964. Why is the separation of user mode and kernel mode considered insufficient for modern security?

- A. Side-channel attacks like Spectre can leak kernel data even from user mode
- B. User mode programs cannot make system calls
- C. Kernel mode does not have access to hardware
- D. There are no privileged instructions in modern CPUs

Answer: A

Q965. What is the multikernel approach to OS design for many-core processors?

- A. It treats the machine as a network of independent cores communicating via messages
- B. It runs a single monolithic kernel replicated on each core
- C. It assigns one core to the kernel and all others to user processes
- D. It uses virtual machines to isolate each core

Answer: A

Q966. What is a container and how does it differ from a virtual machine?

- A. A container shares the host OS kernel while a VM runs its own full OS
- B. A container requires a hypervisor while a VM does not
- C. A container provides stronger isolation than a VM
- D. A container can only run Linux while a VM supports any OS

Answer: A

Q967. What is binary translation in the context of full virtualization?

- A. The hypervisor dynamically rewrites guest privileged instructions at runtime
- B. The compiler converts source code to a different CPU architecture
- C. The OS translates binary files between file formats
- D. The CPU converts binary to decimal during arithmetic operations

Answer: A

Q968. What design challenge does heterogeneous multiprocessing (big.LITTLE) introduce for operating systems?

- A. The scheduler must match task requirements to cores with different performance and power profiles
- B. All cores must run at the same clock speed
- C. The OS must use a separate kernel for each core type
- D. Applications must be compiled separately for each core type

Answer: A

Q969. What is the principle behind a library OS in the context of cloud computing?

- A. Application and OS are compiled together into a single-purpose image for minimal overhead
- B. The OS loads libraries dynamically from a central cloud repository
- C. All OS functions are implemented as shared libraries on a network
- D. Applications use library calls instead of system calls for I/O

Answer: A

Q970. What is the primary purpose of hardware-assisted virtualization extensions like Intel VT-x?

- A. To provide CPU hardware support for efficiently running virtual machines
- B. To increase the clock speed of the CPU
- C. To add more physical cores to the processor
- D. To enable GPUs to execute operating system code

Answer: A

Q971. How does the Darwin kernel used in macOS combine different architectural approaches?

- A. It uses a Mach microkernel core with BSD monolithic services in the same address space
- B. It runs entirely as a microkernel with all services in user space
- C. It uses a pure monolithic design identical to Linux
- D. It relies on a hypervisor to separate kernel components

Answer: A

Q972. What is the difference between a system call and a library call in terms of execution?

- A. A system call transitions to kernel mode while a library call stays in user mode
- B. A library call is always slower than a system call
- C. System calls cannot return values to the caller
- D. Library calls bypass the operating system entirely in all cases

Answer: A

Q973. What is the vDSO mechanism in Linux and why is it used?

- A. It maps kernel code into user space to avoid system call overhead for certain functions
- B. It provides virtual device drivers for emulated hardware
- C. It creates virtual disk storage from RAM
- D. It is a debugging tool for tracing system calls

Answer: A

Q974. What is the challenge of maintaining ABI compatibility in operating system development?

- A. Changes to system call interfaces or data structures can break existing compiled applications
- B. ABI compatibility only affects interpreted languages
- C. Maintaining ABI compatibility slows down hardware performance
- D. ABI compatibility is automatically handled by the CPU

Answer: A

Q975. How do system calls work on x86-64 Linux using the SYSCALL instruction?

- A. The syscall number is placed in RAX and SYSCALL transfers control to the kernel entry point
- B. The CPU automatically determines the system call from the program context
- C. System calls are made by writing to a special memory address
- D. The SYSCALL instruction is only available in kernel mode

Answer: A

Q976. What is the seccomp mechanism in Linux kernel architecture?

- A. A security facility that restricts the system calls a process can make
- B. A tool for compressing the kernel image during boot
- C. A scheduler for secondary storage devices
- D. A method for secure communication between microkernels

Answer: A

Q977. What problem does the Windows Subsystem for Linux (WSL2) solve architecturally?

- A. It runs a real Linux kernel in a lightweight VM to provide native Linux binary compatibility
- B. It translates Linux source code to Windows executables at compile time
- C. It emulates x86 Linux on ARM Windows devices
- D. It replaces the Windows kernel with the Linux kernel

Answer: A

Q978. What is the role of the init system (e.g., systemd) in modern Linux architecture?

- A. It is the first user-space process that initializes services and manages the system lifecycle
- B. It is part of the kernel that handles hardware interrupts
- C. It compiles the kernel from source code during boot
- D. It manages the BIOS firmware settings

Answer: A

Q979. What is the difference between a stable and unstable ABI in kernel module development?

- A. A stable ABI guarantees compatibility across versions while unstable ABI may change at any time
- B. Stable ABI is faster while unstable ABI is more secure
- C. Stable ABI runs in user mode while unstable ABI runs in kernel mode
- D. There is no practical difference in modern operating systems

Answer: A

Q980. How does the Fuchsia OS Zircon kernel differ from traditional Unix-like kernels?

- A. It uses a capability-based object system instead of Unix file descriptors for resource access
- B. It is a monolithic kernel identical to Linux in design
- C. It does not support user-space applications
- D. It runs exclusively on x86 architecture

Answer: A

Q981. What is the double fork technique in Unix and what problem does it solve?

- A. Forking twice makes the grandchild an orphan adopted by init to avoid zombies
- B. It creates two identical copies of a process for load balancing
- C. It forks a process on two different CPUs simultaneously
- D. It creates a backup copy of the process for fault tolerance

Answer: A

Q982. How does the Linux cgroup mechanism control process resource usage?

- A. It groups processes and enforces limits on CPU, memory, and I/O usage per group
- B. It encrypts process data for security
- C. It compresses process memory to save space
- D. It pins processes to specific CPU cores only

Answer: A

Q983. What is the difference between posix_spawn() and fork()+exec() for process creation?

- A. posix_spawn creates a process in a single call avoiding the overhead of copying the parent's address space
- B. posix_spawn only works on Windows systems
- C. fork()+exec() is always faster than posix_spawn
- D. posix_spawn cannot set up file descriptors for the child process

Answer: A

Q984. What are Linux namespaces and how do they relate to process isolation?

- A. They isolate system resources so processes see their own independent view of PIDs, network, and mounts
- B. They provide a naming convention for files on disk
- C. They organize kernel source code into modules
- D. They assign human-readable names to IP addresses

Answer: A

Q985. What is the thundering herd problem in process management?

- A. Multiple waiting processes are all woken up for an event but only one can handle it, wasting resources
- B. Too many processes run simultaneously causing CPU overheating
- C. Processes consume too much memory during startup
- D. Network packets arrive faster than processes can handle them

Answer: A

Q986. How does the Windows process creation model differ from the Unix fork/exec model?

- A. Windows uses CreateProcess() which creates a new process from scratch with a specified executable
- B. Windows uses fork() identically to Unix
- C. Windows cannot create child processes
- D. Windows processes do not have process IDs

Answer: A

Q987. What is process affinity in the context of NUMA architectures?

- A. Binding processes to specific NUMA nodes to keep memory accesses local and reduce latency
- B. Measuring how much a process likes its assigned CPU
- C. Preventing processes from accessing network resources
- D. Grouping processes by the user who created them

Answer: A

Q988. What is the OOM (Out-of-Memory) killer in Linux?

- A. A kernel mechanism that selects and terminates processes when the system runs out of memory
- B. A tool that prevents users from allocating too much disk space
- C. A hardware circuit that shuts down when RAM overheats
- D. A security feature that kills processes attempting buffer overflows

Answer: A

Q989. What is a process descriptor in the Linux kernel and what does it contain?

- A. A task_struct that holds all information the kernel needs about a process
- B. A text description of what the process does
- C. A file containing the process source code
- D. A log of all commands the process has executed

Answer: A

Q990. What is the seccomp-bpf mechanism used for in process sandboxing?

- A. It filters system calls using BPF programs to restrict what a process can do
- B. It encrypts process memory using Berkeley Packet Filters
- C. It monitors network packets sent by a process
- D. It compresses process data to reduce memory usage

Answer: A

Q991. How does the Linux CFS use a red-black tree for scheduling decisions?

- A. Processes are ordered by virtual runtime in the tree, and the leftmost node with smallest vruntime runs next
- B. The tree stores processes sorted by PID and the root node runs next
- C. Processes are randomly inserted and the tree auto-balances priorities
- D. The tree is used only for I/O scheduling, not CPU scheduling

Answer: A

Q992. What is the utilization bound test for rate-monotonic scheduling?

- A. The set of tasks is schedulable if total utilization does not exceed $n(2^{1/n} - 1)$
- B. Tasks are schedulable if each task's period is greater than its execution time
- C. The test checks if total memory usage is below the physical memory limit
- D. Tasks are schedulable only if they all have the same period

Answer: A

Q993. What is the priority inversion problem and how did it affect the Mars Pathfinder?

- A. A high-priority task was blocked by a low-priority task holding a shared resource, causing system resets
- B. The rover's CPU overheated due to excessive computation
- C. A software bug caused the scheduler to run processes in reverse priority order
- D. Network communication delays caused missed commands from Earth

Answer: A

Q994. What is the difference between global scheduling and partitioned scheduling on multicore systems?

- A. Global scheduling uses a single queue for all cores while partitioned assigns each task to a fixed core
- B. Global scheduling is always worse than partitioned scheduling
- C. Partitioned scheduling does not support priority-based algorithms
- D. Global scheduling can only be used with two cores

Answer: A

Q995. What is the BFS (Brain F* Scheduler) design philosophy compared to CFS?**

- A. BFS uses a single global run queue with an earliest eligible virtual deadline algorithm for simplicity
- B. BFS uses per-CPU run queues with work stealing for scalability
- C. BFS is designed only for real-time tasks on embedded systems
- D. BFS uses a multilevel feedback queue identical to traditional Unix schedulers

Answer: A

Q996. How does Earliest Deadline First (EDF) differ from rate-monotonic scheduling for real-time tasks?

- A. EDF assigns dynamic priorities based on deadlines achieving 100% utilization while rate-monotonic uses fixed priorities
- B. EDF only works with aperiodic tasks while rate-monotonic handles periodic tasks
- C. Rate-monotonic achieves higher utilization bounds than EDF
- D. EDF requires hardware support that rate-monotonic does not

Answer: A

Q997. What is work stealing in multiprocessor scheduling?

- A. Idle processors take tasks from the run queues of busy processors to balance load
- B. Processes steal CPU cycles from higher-priority processes
- C. The scheduler removes tasks from slow processors and discards them
- D. Memory is reallocated from one core's cache to another

Answer: A

Q998. What is the SCHED_DEADLINE scheduling class in Linux?

- A. An EDF-based scheduler that guarantees CPU bandwidth to tasks using runtime/deadline/period parameters
- B. A scheduler that assigns deadlines based on process creation time
- C. A file system scheduling class for managing disk deadlines
- D. A network scheduler for prioritizing packet delivery

Answer: A

Q999. What is the problem with using lottery scheduling in practice?

- A. Unpredictable short-term behavior due to randomness even though long-term fairness is achieved
- B. It requires specialized hardware random number generators
- C. It cannot handle more than ten processes simultaneously
- D. It always gives equal time to all processes regardless of priority

Answer: A

Q1000. How does the Energy Aware Scheduler (EAS) in Linux optimize for mobile devices?

- A. It uses CPU energy models to place tasks on the most energy-efficient core that meets performance needs
- B. It shuts down the CPU when battery is below 50%
- C. It reduces screen brightness to save power during computation
- D. It limits the number of processes to reduce energy consumption

Answer: A

Q1001. How does the Linux kernel implement threads differently from traditional UNIX thread implementations?

- A. Linux does not support threads at all
- B. Linux uses the clone() system call to create threads as tasks that share the address space, treating threads and processes uniformly
- C. Linux requires all threads to run in user mode only
- D. Linux threads cannot share file descriptors

Answer: B

Q1002. What is the N:M threading model's primary implementation challenge that led most OSes to abandon it?

- A. It uses too much disk space
- B. Coordinating user-level and kernel-level schedulers is complex and the performance benefits rarely justify the complexity
- C. It cannot support more than two threads
- D. It is incompatible with virtual memory

Answer: B

Q1003. What is thread affinity and why does it matter for performance?

- A. Thread affinity refers to how friendly threads are to each other
- B. It binds a thread to a specific CPU core to exploit warm caches and reduce cache migration overhead
- C. Thread affinity means threads must always run on different cores
- D. It is a security feature that isolates threads

Answer: B

Q1004. What problem does the NUMA-aware thread scheduling try to solve?

- A. Reducing the number of threads in the system
- B. Minimizing memory access latency by scheduling threads on cores close to the memory they access
- C. Preventing threads from accessing shared memory
- D. Converting kernel threads to user threads

Answer: B

Q1005. What is cooperative multithreading and what is its main drawback?

- A. Threads are scheduled by the kernel preemptively; the drawback is high overhead
- B. Threads voluntarily yield the CPU; a misbehaving thread can monopolize the CPU indefinitely
- C. Threads share all variables including the stack; the drawback is data corruption
- D. Only one thread can exist per process; the drawback is no parallelism

Answer: B

Q1006. How do green threads differ from native threads?

- A. Green threads use less electricity
- B. Green threads are scheduled by the runtime or virtual machine in user space, not by the OS kernel
- C. Green threads are always faster than native threads
- D. Green threads can only run on green computing hardware

Answer: B

Q1007. What is the purpose of thread-specific data (TSD) and how does it differ from regular global variables?

- A. TSD is stored on disk for persistence
- B. TSD provides each thread with its own copy of a variable, unlike global variables which are shared among all threads
- C. TSD makes variables immutable across all threads
- D. TSD encrypts thread data for security

Answer: B

Q1008. What are coroutines and how do they relate to threads?

- A. Coroutines are a type of hardware thread
- B. Coroutines are cooperatively scheduled lightweight execution units that can suspend and resume at specific points, often multiplexed over fewer OS threads
- C. Coroutines and threads are identical concepts
- D. Coroutines can only execute sequentially with no concurrency

Answer: B

Q1009. What is the thundering herd problem in the context of thread pools?

- A. Too many threads are created, exhausting memory
- B. Multiple threads are woken up to handle a single event, but only one can process it, wasting resources
- C. Threads run too slowly due to excessive synchronization
- D. All threads in the pool terminate simultaneously

Answer: B

Q1010. What is work stealing in the context of thread pool task scheduling?

- A. A malicious technique where threads steal CPU time
- B. An idle thread takes tasks from the queue of a busy thread to balance the workload dynamically
- C. A method to prevent threads from executing tasks
- D. A technique that forces all threads to execute the same task

Answer: B

Q1011. How does the priority ceiling protocol prevent both deadlock and priority inversion?

- A. By removing all locks from the system
- B. By assigning each lock a priority ceiling equal to the highest priority of any thread that may lock it, and only allowing a thread to acquire a lock if its priority is higher than the ceiling of all currently locked mutexes
- C. By running all threads at the same priority
- D. By disabling interrupts permanently

Answer: B

Q1012. What is sequence consistency (sequential consistency) in the context of concurrent memory operations?

- A. All memory operations execute in parallel without ordering
- B. The result of any execution is the same as if the operations of all processors were executed in some sequential order, preserving each processor's program order
- C. Memory is accessed only sequentially by one processor at a time
- D. It means data is stored in sorted sequence

Answer: B

Q1013. What is the difference between lock-free and wait-free algorithms?

- A. They are identical concepts
- B. Lock-free guarantees system-wide progress (at least one thread progresses); wait-free guarantees per-thread progress (every thread completes in bounded steps)
- C. Wait-free algorithms use more locks than lock-free algorithms
- D. Lock-free means no synchronization is needed

Answer: B

Q1014. How does Read-Copy-Update (RCU) allow readers to access shared data without acquiring locks?

- A. By encrypting the data so readers can safely access it
- B. Readers access the current version without locks; writers create a new copy, update it, and atomically switch the pointer, deferring reclamation of old data until all readers finish
- C. Readers must always copy the data before reading
- D. RCU prevents any concurrent writes

Answer: B

Q1015. What is the ABA problem and how is it typically solved?

- A. It is a problem with alphabetical sorting in concurrent programs
- B. A thread reads value A, another thread changes it to B then back to A; the first thread's compare-and-swap succeeds incorrectly; solved using version counters or hazard pointers
- C. It occurs when three threads simultaneously access the same variable
- D. It is a deadlock pattern involving three resources

Answer: B

Q1016. What is a memory barrier (fence) and why is it necessary on modern processors?

- A. A physical barrier in RAM chips that prevents electrical interference
- B. An instruction that enforces ordering constraints on memory operations, preventing the CPU and compiler from reordering reads and writes across the barrier
- C. A software firewall for memory protection
- D. A method to increase memory bandwidth

Answer: B

Q1017. How does transactional memory simplify concurrent programming compared to lock-based approaches?

- A. It eliminates the need for any concurrency control
- B. Programmers designate code regions as transactions that execute atomically; conflicts are detected automatically, eliminating manual lock management and reducing deadlock risk
- C. It makes all memory accesses sequential
- D. It converts concurrent programs into single-threaded programs

Answer: B

Q1018. In the context of Linux kernel synchronization, what are per-CPU variables and why are they efficient?

- A. Variables that count the number of CPUs in the system
- B. Each CPU has its own copy of the variable, eliminating the need for locking since no sharing occurs between CPUs
- C. Variables that are shared equally among all CPUs
- D. Variables that store CPU temperature readings

Answer: B

Q1019. What is the difference between optimistic and pessimistic concurrency control?

- A. Optimistic control is always faster
- B. Pessimistic control acquires locks before accessing data; optimistic control proceeds without locks and validates at commit time, aborting on conflict
- C. They produce different results for the same operations
- D. Optimistic control does not support concurrent access

Answer: B

Q1020. What is the sleeping barber problem and what synchronization challenge does it illustrate?

- A. A problem about scheduling CPU sleep states
- B. A barber sleeps when no customers are present and customers must wake the barber or wait; it illustrates coordination between producers and consumers with a bounded waiting area
- C. A deadlock scenario involving three barber threads
- D. A memory management problem involving page sleeping

Answer: B

Q1021. What is the wait-die scheme for deadlock prevention in database systems?

- A. Processes wait indefinitely until resources become available
- B. An older process waits for a younger one to release a resource, but a younger process requesting a resource held by an older one is aborted (dies)
- C. All processes die when a deadlock is detected
- D. Processes are assigned random wait times

Answer: B

Q1022. How does the edge-chasing algorithm detect deadlocks in distributed systems?

- A. By checking all resource allocation graphs simultaneously
- B. Probe messages are forwarded along edges of the distributed wait-for graph; if a probe returns to its initiator, a cycle (deadlock) exists
- C. By centralizing all resource information on one node
- D. By terminating all processes on all nodes periodically

Answer: B

Q1023. What is a phantom deadlock in distributed deadlock detection?

- A. A deadlock that occurs in virtual machines only
- B. A falsely detected deadlock caused by delays in propagating status information across nodes in a distributed system
- C. A deadlock involving phantom processes that do not exist
- D. A deadlock that occurs during system startup

Answer: B

Q1024. What is the relationship between deadlock and livelock in resource contention?

- A. Livelock is a more severe form of deadlock
- B. In deadlock, processes are blocked and make no progress; in livelock, processes continuously change state in response to each other but still make no progress
- C. Livelock is identical to deadlock
- D. Livelock occurs only with non-preemptive scheduling

Answer: B

Q1025. How does lock ordering (hierarchical locking) prevent deadlock in practice?

- A. By requiring all locks to be held for the same duration
- B. By defining a global order on all locks and requiring every thread to acquire locks only in that order, preventing cycles in the wait-for graph
- C. By limiting the number of locks in the system to one
- D. By making all locks recursive

Answer: B

Q1026. What is the knot in a resource allocation graph?

- A. A cycle involving exactly two processes
- B. A set of nodes such that every node in the set has a path to every other node in the set with no outgoing edges leaving the set
- C. A single node with no edges
- D. An edge that connects a resource to itself

Answer: B

Q1027. What is two-phase locking and how does it relate to deadlock in database systems?

- A. It prevents deadlock by ensuring locks are acquired in two distinct phases
- B. In the growing phase all needed locks are acquired and in the shrinking phase locks are released; it ensures serializability but does not prevent deadlock
- C. It acquires and releases each lock twice for safety
- D. It uses two separate lock managers to prevent deadlock

Answer: B

Q1028. How does the Linux kernel detect potential deadlocks using lockdep?

- A. By running the Banker's Algorithm at each lock acquisition
- B. By tracking lock acquisition order at runtime and checking for potential cycles in the lock dependency graph, even if the actual deadlock hasn't occurred
- C. By limiting the number of locks to 10
- D. By preventing any two threads from running simultaneously

Answer: B

Q1029. What is the convoy effect in the context of resource scheduling, and how does it relate to deadlock-like behavior?

- A. A high-speed scheduling algorithm
- B. Many processes queue behind a slow process holding a resource, causing a chain of delays similar to but distinct from deadlock
- C. All processes execute in parallel without any delays
- D. Processes are permanently blocked and never execute

Answer: B

Q1030. What is the difference between centralized and distributed deadlock detection, and what are the trade-offs?

- A. They are identical approaches applied to different operating systems
- B. Centralized detection maintains a global wait-for graph at one coordinator (simpler but single point of failure); distributed detection uses local algorithms with message passing (more resilient but may detect phantom deadlocks)
- C. Centralized detection is always faster and more accurate
- D. Distributed detection eliminates all false positives

Answer: B

Q1031. How does the clock (second-chance) algorithm approximate LRU replacement?

- A. It uses a hardware timer to track page usage
- B. Pages are arranged in a circular list with a reference bit; the algorithm scans for a page with reference bit 0, clearing bits as it goes, giving referenced pages a second chance
- C. It randomly selects a page for replacement
- D. It replaces the page closest to the clock's 12 o'clock position

Answer: B

Q1032. What is copy-on-write (COW) optimization in fork() and how does it work?

- A. It copies all pages immediately when fork() is called
- B. Parent and child initially share the same physical pages marked read-only; a page is copied only when either process writes to it, reducing unnecessary copying
- C. It prevents child processes from writing to any memory
- D. It copies memory from disk to RAM on write operations

Answer: B

Q1033. What is Address Space Layout Randomization (ASLR) and how does it improve security?

- A. It sorts addresses in alphabetical order
- B. It randomizes the memory layout of a process (stack, heap, libraries) at each execution, making buffer overflow exploits harder by unpredictable addresses
- C. It adds encryption to all memory addresses
- D. It assigns the same layout to all processes for consistency

Answer: B

Q1034. How does the buddy system allocator handle memory fragmentation?

- A. It eliminates fragmentation completely
- B. It splits memory blocks into power-of-two sized buddies and merges adjacent free buddies back together, reducing external fragmentation but causing internal fragmentation
- C. It uses a linked list of arbitrary block sizes
- D. It allocates memory in a single contiguous block

Answer: B

Q1035. What is the slab allocator and why is it efficient for kernel memory allocation?

- A. A disk allocation algorithm for solid-state drives
- B. It pre-allocates caches of objects of specific sizes from contiguous memory slabs, eliminating fragmentation and initialization overhead for frequently allocated kernel objects
- C. A memory compaction technique for user programs
- D. A page replacement algorithm for virtual memory

Answer: B

Q1036. What is page coloring and how does it improve cache performance?

- A. It assigns colors to pages for visual debugging
- B. It assigns physical frames to virtual pages such that pages mapping to the same cache set are distributed across frames, reducing cache conflicts
- C. It marks pages as read-only using color codes
- D. It groups pages by their access frequency

Answer: B

Q1037. How do huge pages (large pages) improve TLB performance?

- A. They reduce the size of each page
- B. Fewer TLB entries cover the same amount of memory, reducing TLB misses for applications with large memory footprints
- C. They increase the number of TLB entries available
- D. They bypass the TLB entirely

Answer: B

Q1038. What is memory-mapped file I/O and what advantage does it offer?

- A. A technique that stores files in CPU registers
- B. It maps a file's contents into a process's virtual address space, allowing file access through memory operations instead of read/write system calls, reducing system call overhead
- C. It encrypts files during I/O operations
- D. It stores all file metadata in physical memory

Answer: B

Q1039. What is the page-fault frequency (PFF) approach to thrashing prevention?

- A. It sets a minimum page fault rate for each process
- B. It monitors each process's page fault rate; if too high, more frames are allocated; if too low, frames are reclaimed, dynamically adjusting allocation
- C. It prevents all page faults by loading every page into memory
- D. It increases the swap space size when faults are frequent

Answer: B

Q1040. What is Kernel Same-page Merging (KSM) and where is it most beneficial?

- A. A technique to merge kernel code pages with user code pages
- B. It scans memory for identical pages across processes and merges them into a single shared copy (with COW), most beneficial in virtualization where VMs run similar OSes
- C. A method to compress kernel memory pages
- D. A page replacement algorithm for kernel pages

Answer: B

Q1041. How does the log-structured file system (LFS) approach differ from traditional file systems?

- A. LFS stores all files in a single log file
- B. LFS writes all changes (data and metadata) sequentially to a log, converting random writes to sequential writes for better write performance, using a cleaner for garbage collection
- C. LFS reads files from the end to the beginning
- D. LFS does not support directories

Answer: B

Q1042. How does ZFS ensure end-to-end data integrity?

- A. By using parity bits only
- B. By storing checksums in parent blocks (not with the data block itself), detecting silent data corruption, bit rot, and phantom writes
- C. By encrypting all data with AES
- D. By writing data twice to the same location

Answer: B

Q1043. What is FUSE and why is it significant for file system development?

- A. A hardware component for fast USB storage
- B. A framework that allows implementing file systems in user space without modifying the kernel, simplifying development and enabling novel file systems
- C. A file compression algorithm
- D. A network protocol for distributed file systems

Answer: B

Q1044. How do extents improve upon traditional block pointers in file systems like ext4?

- A. Extents store data in smaller units for precision
- B. An extent describes a contiguous range of blocks with a single start-block and length, dramatically reducing metadata overhead for large files
- C. Extents eliminate the need for inodes
- D. Extents only work with SSDs

Answer: B

Q1045. What is a copy-on-write file system and what advantages does it offer?

- A. A file system that copies files on every read operation
- B. Instead of overwriting data in place, new data is written to a new location and metadata is updated atomically, enabling snapshots, preventing partial writes, and maintaining consistency
- C. A file system that prevents all write operations
- D. A file system that duplicates the entire disk on each write

Answer: B

Q1046. How does the B+ tree structure benefit NTFS and other modern file systems?

- A. B+ trees are slower but use less memory
- B. B+ trees provide $O(\log n)$ lookup, insertion, and deletion for directory entries and file extents, with all data in leaf nodes for efficient range queries
- C. B+ trees only store file names, not their locations
- D. B+ trees are only used for binary files

Answer: B

Q1047. What is the TRIM command and why is it essential for SSD-backed file systems?

- A. It compresses files to save SSD space
- B. It informs the SSD which blocks are no longer in use, allowing the SSD's FTL to erase and reclaim those blocks proactively for better write performance
- C. It trims file names to save directory space
- D. It limits the number of files per directory

Answer: B

Q1048. What is deduplication in enterprise file systems and what are its trade-offs?

- A. It duplicates data for redundancy
- B. It identifies and eliminates duplicate data blocks, storing only unique blocks with references, saving storage but adding CPU and memory overhead for hash computation
- C. It removes all files older than 30 days
- D. It deletes duplicate file names only

Answer: B

Q1049. How does the Network File System (NFS) handle stateless server design and what are the implications?

- A. NFS requires the server to maintain all client state
- B. The NFS server does not track which clients have opened files; each request is self-contained, simplifying crash recovery but requiring idempotent operations
- C. NFS does not support any crash recovery
- D. NFS stores all state on the client side only

Answer: B

Q1050. What is the purpose of file system snapshots and how are they efficiently implemented in COW file systems?

- A. Snapshots are photographs of the server hardware
- B. Snapshots capture the file system state at a point in time; in COW file systems, they are nearly instant because only the root metadata block is copied, sharing unchanged blocks with the live file system
- C. Snapshots require copying the entire disk
- D. Snapshots delete all files created after a specific date

Answer: B

Q1051. What is write amplification in SSDs and why is it a problem?

- A. Writing louder signals to flash cells for reliability
- B. The actual amount of data written to flash exceeds the logical data written by the host due to garbage collection and wear leveling, reducing SSD lifespan and performance
- C. Writing data to multiple SSDs simultaneously
- D. Increasing the write speed beyond the SSD's rated capacity

Answer: B

Q1052. What is the Flash Translation Layer (FTL) in SSDs?

- A. A physical layer of insulation in the SSD
- B. Firmware that maps logical block addresses to physical flash locations, managing wear leveling, garbage collection, and bad block handling
- C. A network protocol for flash storage
- D. A file system designed specifically for flash memory

Answer: B

Q1053. How does NVMe improve upon the AHCI interface for SSDs?

- A. NVMe uses serial communication while AHCI uses parallel
- B. NVMe supports up to 65,535 I/O queues with 65,536 entries each and uses direct PCIe connection, dramatically reducing latency and increasing throughput compared to AHCI's single queue of 32 entries
- C. NVMe is designed for hard disk drives only
- D. AHCI is faster than NVMe for random access

Answer: B

Q1054. What is shingled magnetic recording (SMR) in HDDs?

- A. A technique using shingle-shaped disk platters
- B. Overlapping write tracks like roof shingles to increase data density, but requiring sequential writes within zones since writing a track partially overwrites adjacent tracks
- C. A method of reading data from multiple tracks simultaneously
- D. A technique to reduce disk noise

Answer: B

Q1055. What is NCQ (Native Command Queuing) in SATA drives?

- A. A method to queue network commands for remote drives
- B. A feature allowing the drive to internally reorder queued I/O commands to minimize seek and rotational delays, improving throughput
- C. A technique that queues commands in the OS driver only
- D. A command that queries the native format of the disk

Answer: B

Q1056. How does RAID 6 improve fault tolerance over RAID 5?

- A. RAID 6 uses triple mirroring
- B. RAID 6 uses two independent parity calculations (P and Q), tolerating two simultaneous disk failures instead of one
- C. RAID 6 has no parity overhead
- D. RAID 6 stripes data without any redundancy

Answer: B

Q1057. What is the NOOP (No Operation) I/O scheduler and when is it appropriate?

- A. It performs no I/O operations at all
- B. It implements a simple FIFO queue with request merging only, appropriate for SSDs and virtualized environments where the device or hypervisor handles scheduling
- C. It is a disk scheduling algorithm for maximum throughput on HDDs
- D. It disables the disk entirely

Answer: B

Q1058. What are Zoned Namespaces (ZNS) in NVMe SSDs and how do they differ from conventional SSDs?

- A. ZNS SSDs have different geographic zones for data storage
- B. ZNS exposes the SSD's internal zone structure to the host, requiring sequential writes within zones but eliminating the need for the FTL's garbage collection, reducing write amplification
- C. ZNS is a security zoning feature for data isolation
- D. ZNS SSDs have larger physical size than conventional SSDs

Answer: B

Q1059. What is the Advanced Format (4K sector) standard and why was it adopted?

- A. A format that stores data in 4KB RAM pages
- B. It increased the physical sector size from 512 bytes to 4096 bytes, improving error correction efficiency, format efficiency, and storage density on modern drives
- C. A new formatting standard that only applies to SSDs
- D. A technique that formats disks four times faster

Answer: B

Q1060. What is scatter-gather DMA and how does it improve I/O performance?

- A. It scatters data across multiple disks randomly
- B. It allows DMA to transfer data between non-contiguous memory regions and a device in a single operation using a list of memory descriptors, avoiding extra copy operations
- C. It gathers interrupts from multiple devices into one
- D. It distributes I/O requests across multiple CPUs

Answer: B

Q1061. How does Linux epoll improve over select() and poll() for I/O multiplexing?

- A. epoll uses more memory than select()
- B. epoll uses an event-driven model that only reports changes, avoids scanning all file descriptors, and scales $O(1)$ with the number of monitored descriptors
- C. epoll works only with network sockets
- D. select() is always faster than epoll

Answer: B

Q1062. What is the IOMMU and how does it enhance I/O security and functionality?

- A. A memory unit for storing I/O device firmware
- B. Hardware that translates I/O device virtual addresses to physical addresses, preventing devices from accessing unauthorized memory regions and enabling DMA for virtualized guests
- C. A specialized I/O processor that replaces the CPU
- D. A memory management unit within the I/O device itself

Answer: B

Q1063. What is the difference between edge-triggered and level-triggered interrupt notifications?

- A. Edge-triggered uses higher voltage levels
- B. Edge-triggered fires on the transition (change) of the signal; level-triggered fires as long as the signal is asserted, requiring different handling strategies to avoid missed or spurious interrupts
- C. Level-triggered is always more efficient
- D. Edge-triggered is only used for software interrupts

Answer: B

Q1064. What is io_uring in modern Linux and why is it significant?

- A. A circular buffer for storing I/O errors
- B. An asynchronous I/O interface using shared ring buffers between user space and kernel, minimizing system call overhead and enabling high-performance I/O without context switches
- C. A debugging tool for I/O operations
- D. A type of network ring topology

Answer: B

Q1065. What is zero-copy I/O and how does sendfile() implement it?

- A. I/O where no data is actually transferred
- B. Data is transferred from disk to network socket directly through kernel buffers without being copied to user space, reducing CPU usage and memory bandwidth consumption
- C. A technique that compresses data to zero bytes during transfer
- D. I/O that creates zero backup copies

Answer: B

Q1066. What is interrupt coalescing and why is it used in high-speed network adapters?

- A. Combining multiple interrupt controllers into one
- B. Delaying and batching multiple interrupts into fewer interrupt notifications to reduce per-packet interrupt overhead at high data rates
- C. Converting interrupts into polling operations
- D. Encrypting interrupt signals for security

Answer: B

Q1067. What is RDMA (Remote Direct Memory Access) and how does it bypass the traditional I/O stack?

- A. A technique to access RAM on a remote machine through SSH
- B. It allows one computer to directly read from or write to another computer's memory without involving either CPU or OS, bypassing the kernel network stack entirely
- C. A remote debugging protocol for memory analysis
- D. A technique to access disk drives over the network

Answer: B

Q1068. What is the SPDK (Storage Performance Development Kit) approach to storage I/O?

- A. A kernel module for legacy storage devices
- B. It moves storage drivers into user space, uses polled mode instead of interrupts, and provides lockless designs to eliminate kernel overhead and achieve near-hardware-level NVMe performance
- C. A standard development kit for building storage hardware
- D. A graphical tool for monitoring storage performance

Answer: B

Q1069. How does virtio enable efficient I/O for virtual machines?

- A. By providing dedicated physical hardware to each VM
- B. It defines a standardized interface between VMs and the hypervisor using shared memory ring buffers (virtqueues), avoiding the overhead of emulating real hardware devices
- C. By running all VMs on the same physical I/O device
- D. By disabling I/O for virtual machines entirely

Answer: B

Q1070. How does return-oriented programming (ROP) bypass Data Execution Prevention (DEP/NX)?

- A. By disabling the CPU's NX bit
- B. By chaining together existing code sequences (gadgets) ending in return instructions from legitimate libraries, executing arbitrary computations without injecting new code
- C. By encrypting the malicious code so DEP cannot detect it
- D. By running code in kernel mode where DEP does not apply

Answer: B

Q1071. What is the Bell-LaPadula security model and what property does it enforce?

- A. A model that allows all users to read and write all data
- B. A mandatory access control model enforcing 'no read up, no write down' to prevent information flow from higher to lower security levels
- C. A model based on role-based access control
- D. A model that enforces data integrity but not confidentiality

Answer: B

Q1072. How does SELinux implement mandatory access control differently from traditional Unix permissions?

- A. SELinux replaces all file permissions with encryption
- B. SELinux assigns security labels to all subjects and objects and enforces policy rules for every access decision, regardless of traditional DAC permissions
- C. SELinux only adds network security controls
- D. SELinux uses the same permission model as standard Unix

Answer: B

Q1073. What is the confused deputy problem in computer security?

- A. When a system administrator is confused about access policies
- B. When a privileged program is tricked by a less-privileged entity into misusing its authority to perform unauthorized actions on the attacker's behalf
- C. When two processes have the same security level and confuse each other
- D. When encryption keys are mixed up between users

Answer: B

Q1074. What is control flow integrity (CFI) and how does it protect against code-reuse attacks?

- A. It controls the flow of network traffic
- B. It restricts the program's control flow transfers to only those targets determined at compile time, preventing ROP and JOP attacks from diverting execution
- C. It encrypts the program's control flow graph
- D. It monitors data flow but not control flow

Answer: B

Q1075. What is the Trusted Platform Module (TPM) and what security functions does it provide?

- A. A software-only security module for web browsers
- B. A dedicated hardware chip that provides secure key generation, storage, platform integrity measurement, and remote attestation capabilities
- C. A removable storage device for encrypted backups
- D. A network firewall appliance

Answer: B

Q1076. How does Memory Tagging Extension (MTE) help prevent memory safety vulnerabilities?

- A. It adds labels to memory for human readability
- B. It associates a 4-bit tag with each memory granule and each pointer; access is only permitted when the pointer's tag matches the memory's tag, detecting use-after-free and buffer overflows
- C. It encrypts all memory with unique tags
- D. It compresses memory to prevent overflow

Answer: B

Q1077. What is a zero-day vulnerability and why is it particularly dangerous?

- A. A vulnerability that takes zero days to fix
- B. A previously unknown vulnerability being exploited before the vendor has released a patch, leaving all systems defenseless
- C. A vulnerability that only exists for one day
- D. A vulnerability in programs that are zero days old

Answer: B

Q1078. What is attestation in the context of trusted computing?

- A. A user testifying in court about a security breach
- B. A process by which a system cryptographically proves its hardware and software configuration to a remote verifier, establishing trust in its integrity
- C. A method of authenticating users via attestation letters
- D. A technique for attesting to the speed of encryption

Answer: B

Q1079. How does the Biba integrity model differ from the Bell-LaPadula model?

- A. The Biba model is a more recent version of Bell-LaPadula
- B. Bell-LaPadula focuses on confidentiality (no read up, no write down); Biba focuses on integrity (no read down, no write up), preventing corruption of higher-integrity data
- C. Biba and Bell-LaPadula are identical models
- D. Biba allows unrestricted reading and writing

Answer: B

Q1080. What is over-provisioning in SSDs and why is it necessary?

- A. Providing more SSDs than needed as backups
- B. Reserving a percentage of flash capacity as spare blocks for garbage collection, wear leveling, and bad block replacement, improving sustained write performance and endurance
- C. Writing more data than the SSD can hold
- D. Providing extra power supply to the SSD

Answer: B