WAYGROUND formerly Quizizz **Worksheets**

OS UNIT-1 PVPSIT

Total questions: 80

Worksheet time: 40mins

Instructor name: Mr. PRASHANT ATMAKURI

Name

Class

Date

1. Which statement best captures one primary purpose of an operating system?

   a) Increase raw processor clock frequency
   b) Transform compiled code into machine hardware
   c) Provide a convenient program execution environment
   d) Eliminate all user errors automatically

2. What role does an operating system play in resource allocation?

   a) Compiles programs into binary executables
   b) Schedules network packets across routers
   c) Encrypts all stored data by default
   d) Distributes hardware resources fairly and efficiently

3. Which control function is commonly attributed to an operating system?

   a) Supervision of program execution to prevent errors
   b) Dynamic overclocking of the CPU cores
   c) Guaranteed elimination of I/O latency
   d) Automatic debugging of all source code

4. Why might a GUI be considered to “waste” CPU cycles on a single-user system?

   a) It prioritizes user interaction over raw utilization
   b) It blocks access to system memory permanently
   c) It forces all programs to run in kernel mode
   d) It disables multitasking on modern processors

5. In a single-user system, what is the recommended optimization goal for the operating system?

   a) Maximize server throughput
   b) Maximize compiler optimization
   c) Maximize usability for the user
   d) Minimize network jitter

6. Which statement best explains why a system that appears to “waste” resources might not be wasteful?

   a) All resources should remain idle to prevent wear
   b) Improved interaction can enhance overall efficiency
   c) Higher CPU usage guarantees faster performance
   d) Unused cycles always reduce battery drain

7. What is the main difficulty in designing a real-time operating system?

   a) Providing perfect graphical fidelity
   b) Supporting unlimited concurrent users
   c) Meeting fixed time constraints for tasks
   d) Achieving the highest benchmark scores

8. If a real-time task misses its deadline, what is a likely consequence for the system?

   a) Breakdown of the entire system behavior
   b) Compilation of tasks into faster binaries
   c) Automatic rescheduling with no impact
   d) Increased screen resolution for graphics

9. Which pair of functions together describes OS control responsibilities mentioned in the material?

   a) Packet routing and network cable testing
   b) Battery calibration and power-supply design
   c) Application design and code refactoring
   d) Supervision of execution and I/O device management

10. Which scenario best justifies allowing apparent resource “waste” on a desktop OS?

    a) Enabling rich GUI features to improve user experience
    b) Running all tasks in real mode for simplicity
    c) Disabling virtual memory to reduce disk usage
    d) Reducing thread counts to avoid context switches

11. Which statement best describes why kernel mode enforces protection of critical resources?

    a) User mode has larger memory than kernel mode
    b) Kernel mode disables all device communication
    c) User mode lacks access to certain CPU instructions
    d) Kernel mode runs slower than user mode for safety

12. Which operation is typically restricted to kernel mode to protect hardware devices?

a) Issuing a user-level function call
b) Accessing an I/O device
c) Starting an application process
d) Reading the wall-clock time

13. Which of the following is a privileged instruction that should not execute in user mode?

a) Issue a trap instruction
b) Modify entries in device-status table
c) Read the clock
d) Switch from kernel to user mode

14. What is the primary reason interrupts can be enabled or disabled only in kernel mode?

a) Because hardware cannot signal in user mode
b) To save battery on mobile devices
c) To prevent untrusted code from blocking critical events
d) Because user programs always run faster

15. Which operation can be safely performed in user mode?

a) Clear memory globally
b) Set value of timer
c) Turn off interrupts
d) Issue a trap instruction

16. Why might embedding popular applications inside an operating system be argued against?

a) It introduces security vulnerabilities and bloat
b) It prevents kernel-mode execution entirely
c) It reduces memory fragmentation only
d) It guarantees better performance for all apps

17. Which capability difference between kernel and user mode most limits what CPU can do in user mode?

a) Higher cache latency in user mode
b) Different compiler optimizations
c) Larger virtual address space in kernel mode
d) Restricted execution of certain instructions

18. Choose the list that contains only privileged operations.

a) Increment variable, branch, call function
b) Read clock, issue trap, start thread
c) Open file, print text, create socket
d) Set timer, clear memory, access I/O

19. If a user-mode program could disable interrupts, what risk would most likely occur?

a) Cache would become permanently incoherent
b) Virtual memory would stop mapping pages
c) Files would always open read-only
d) System could miss critical device events

20. Which design benefit is cited in favor of including applications within the operating system?

a) Simpler distinction between modes
b) Better access to kernel features for performance
c) Guaranteed elimination of security flaws
d) Automatic hardware compatibility

21. Which difficulty arises when an operating system stores sensitive data in unprotected memory while using a protected OS partition?

a) Unauthorized access to OS data becomes possible
b) Cache lines evict faster components' data
c) User mode cannot call device drivers directly
d) Kernel timers fail due to lost interrupts

22. What is one possible use of CPUs that support more than two modes of operation?

a) Provide finer-grained security among user groups
b) Eliminate the need for kernel and user modes
c) Guarantee real-time scheduling for all threads
d) Increase disk cache size beyond main memory

23. A CPU has multiple user modes distinguishing groups. What capability could this enable?

a) USB devices bypass all driver software
b) Caches automatically synchronize component clocks
c) Users in the same group execute each other's code
d) Timers never need periodic interrupts

24. How could USB device drivers benefit from an additional CPU mode besides user and kernel?

a) Replace hardware interrupts with polling
b) Avoid consistency between cache and components
c) Store passwords inside device firmware
d) Run in a quasi-user/kernel mode for service

25. Which approach can a program use to compute current time using timer interrupts?

a) Set a future timer, sleep, update state on interrupt
b) Use cache size to store the latest clock ticks
c) Query disk controller for timestamp values
d) Continuously busy-wait and count loop iterations

26. When using timer interrupts to track time, what repeated process helps maintain accuracy?

a) Switch USB drivers to kernel-only execution
b) Continually set interrupts and update local state
c) Increase CPU modes until drift is eliminated
d) Flush caches each time data is exchanged

27. Why are caches useful when two components transfer data at differing speeds?

a) They eliminate the need for interrupts entirely
b) They buffer with intermediate speed between components
c) They move operating system into protected memory
d) They convert all devices to identical clock rates

28. What problem do caches help solve in multi-component systems?

a) Preventing USB devices from loading drivers
b) Creating new CPU execution privilege levels
c) Bridging speed mismatch during data transfers
d) Guaranteeing monotonic timer scheduling

29. What is a key requirement for data stored in a cache relative to component data?

a) Remain longer than all disk blocks
b) Use only user-mode read operations
c) Be encrypted with kernel-only keys
d) Maintain consistency with component data

30. If a cache were made as large as the device it caches, what remains a concern?

a) Eliminating all timer interrupts forever
b) Reducing CPU modes to a single level
c) Running drivers entirely in user space
d) Keeping cached data consistent with device

31. Which statement best characterizes the client-server model in distributed systems?

a) Nodes act as peers with equal responsibilities
b) Any node may serve or request for others
c) Clients request services that servers provide
d) Roles are fluid without strict separation

32. In a peer-to-peer model, how are the roles of nodes typically defined?

a) Strictly as clients and servers only
b) Only servers provide services to clients
c) As peers that may act as client or server
d) Clients cannot provide services to others

33. Which scenario best illustrates client-server behavior in the recipe-sharing example?

a) Recipes stored on every node equally
b) A node requests a recipe from any peer node
c) All recipes stored on a central server node
d) Peers exchange recipes without a central role

34. What is a key property of peer nodes in the peer-to-peer recipe system?

a) They always forward requests to a server
b) They may both request and provide recipes
c) They can only store local recipes
d) They never communicate directly with peers

35. Which issue arises when a cached datum changes in a multiprocessor system?

a) All caches must be invalidated instantly
b) No update is needed anywhere
c) Only main memory needs modification
d) The cache containing the datum must be updated

36. Under what conditions can a component be eliminated by an equal-sized cache?

a) Cache is larger but slower than component
b) Cache is cheaper but volatile only
c) Component retains data only when powered
d) Cache and component have equal state-saving capacity

37. Why is cache retention after power removal relevant?

a) It ensures data persists without electricity
b) It speeds up cache write operations
c) It increases network throughput
d) It lowers the cost of faster storage

38. What affordability trade-off is noted regarding caches?

a) Cheaper caches are always faster
b) Equal-sized caches are universally economical
c) Slow storage tends to be unaffordable
d) Faster storage tends to be more expensive

39. In peer-to-peer systems, what may a node do besides requesting a service?

a) Provide a service to other peers
b) Act only as a client forever
c) Reject all incoming peer requests
d) Forward requests to a central server

40. What distinction is emphasized in the client-server model?

a) Strict roles separating client and server
b) Random selection of service providers
c) Decentralized storage of all data
d) Fluid node roles without separation

41. What is the primary purpose of system calls in an operating system?

a) Encrypt inter-process messages for security
b) Allow user processes to request OS services
c) Provide device drivers direct hardware control
d) Schedule processes using preemptive algorithms

42. Which statement best describes the role of a command interpreter (shell)?

a) Reads commands and invokes system calls
b) Maintains the process table entries
c) Implements virtual memory paging
d) Compiles kernel modules dynamically

43. Why is the command interpreter usually separate from the kernel?

a) It stores user passwords securely
b) It must run in privileged kernel mode
c) It changes frequently compared to kernel
d) It requires hardware interrupts to execute

44. On UNIX systems, which pair of system calls starts a new process from a shell?

a) fork() followed by exec()
b) spawn() followed by wait()
c) open() followed by read()
d) init() followed by kill()

45. What does fork() do when invoked by a running process?

a) Terminates the parent process immediately
b) Overlays memory with a new executable
c) Allocates a new kernel thread only
d) Clones the current executing process

46. What is the effect of exec() after a successful fork()?

a) Creates an additional child process copy
b) Replaces the process image with executable
c) Flushes all open file descriptors
d) Switches the process to kernel-only mode

47. How are system programs characterized in relation to system calls?

a) Independent of kernel interfaces
b) Primarily graphics libraries
c) Exclusive to device manufacturers
d) Bundles of useful system calls

48. What benefit do system programs provide to users?

a) Automatic recovery from kernel panics
b) Basic functionality without writing code
c) Guaranteed real-time performance
d) Direct access to hardware registers

49. What is a main advantage of a layered OS design?

a) Faster context switching overall
b) Eliminates need for system calls
c) Easier debugging and modification
d) Guarantees zero latency interrupts

50. Which is a potential disadvantage of layered OS design?

a) Potential performance overhead across layers
b) Lack of modularity in components
c) Inability to port between architectures
d) No support for multiuser environments

51. Which operating-system service ensures a program’s code is loaded into memory and begins running?

a) I/O operations
b) File manipulation
c) Program execution
d) Communications
e) Error detection

52. Which service manages low-level interactions with disks, tapes, and devices through controller-specific commands?

a) Communications
b) Program execution
c) Error detection
d) I/O operations
e) File manipulation

53. Which service handles creation, deletion, naming, allocation, and deallocation of files and blocks?

a) I/O operations
b) Communications
c) File manipulation
d) Error detection
e) Program execution

54. Which service turns messages into packets, sends them over a medium, and reassembles them at the destination?

a) Error detection
b) Communications
c) I/O operations
d) Program execution
e) File manipulation

55. At which levels does error detection occur according to the described operating-system service?

a) Compiler and linker
b) Network only
c) Firmware only
d) Hardware and software
e) Application only

56. Why can’t user-level programs be trusted to allocate CPU time properly during program start?

a) They detect all hardware faults
b) They reorder packets
c) They have excess memory
d) They cannot create files
e) They lack scheduling control

57. Which statement best explains why user programs should not access devices directly?

a) They eliminate packet loss
b) They always run faster
c) They might bypass protection
d) They guarantee integrity checks
e) They cannot open files

58. During file deletion, which step ensures previously allocated blocks become available?

a) Formatting controller tapes
b) Reordering network packets
c) Caching device registers
d) Allocating CPU quanta
e) Deallocating file blocks

59. Which duty in communications requires coordination to avoid packets meant for other processes?

a) File renaming policy
b) Device access control
c) Disk head scheduling
d) CPU register saving
e) Media consistency checks

60. What global responsibility must the operating system perform when errors are process-independent, such as disk corruption?

a) User-level packet routing
b) Centralized error handling
c) Application-managed storage
d) Direct device programming
e) Per-file CPU allocation

61. Which scenario most justifies storing an operating system in firmware rather than on disk?

a) Embedded device without a usable file system
b) Server using network-attached storage volumes
c) Laptop with multiple hard drives installed
d) Desktop with a large RAID array configured

62. What is the primary role of a boot manager during system startup?

a) Scan hardware devices for drivers
b) Load user applications into main memory
c) Encrypt the boot sector for security
d) Select which operating system to boot

63. Why might a system default to booting a specific operating system if the user makes no selection?

a) Firmware cannot display any boot menu
b) Operating systems require manual confirmation
c) File systems always mark one partition active
d) Boot managers are designed with a default

64. Where are boot managers typically stored to be recognized during startup?

a) Inside the CPU instruction cache
b) On removable media like USB sticks
c) At fixed locations on the hard disk
d) In the GPU firmware memory

65. If a machine runs Windows and several Linux distributions, what enables choosing among them at boot time?

a) A boot manager presenting selections
b) A scheduler allocating boot priorities
c) A hypervisor switching kernels dynamically
d) A compiler generating multi-boot binaries

66. Which statement best explains storing an OS on disk for general-purpose computers?

a) CPUs cannot execute firmware images
b) Networks require disk-backed paging
c) Disk storage allows flexible file systems
d) Firmware always lacks address space

67. A system must boot into a default OS unless the user intervenes. Which design choice ensures this behavior?

a) Disable all non-default boot entries
b) Place the default kernel in CPU registers
c) Set the boot manager timeout and default
d) Configure firmware to bypass disk probes

68. During the bootstrap process, which component initially runs before any operating system is loaded?

a) User-level shell interface
b) Boot manager program
c) Filesystem journaling module
d) Virtual memory subsystem

69. An embedded device lacks persistent disk storage. Which boot strategy is most appropriate?

a) Install a boot manager on a DVD
b) Compress the OS into swap space
c) Store the OS image within firmware
d) Mount a network filesystem at boot

70. Which design feature helps a boot manager recognize installed operating systems on startup?

a) Filesystem journaling scanning application caches
b) GPU drivers enumerating kernels automatically
c) CPU microcode mapping partitions to processes
d) Operating systems stored at known locations

71. Which component manages process scheduling and resource allocation in a modern operating system?

a) Bootloader stage manager
b) Window compositor module
c) File system driver layer
d) Kernel and scheduler core

72. In CPU architecture, which unit primarily handles arithmetic and logical operations on integer data?

a) Branch prediction unit
b) Arithmetic Logic Unit
c) Instruction decode queue
d) Floating-point pipeline

73. What is the main purpose of virtual memory in a computer system?

a) Encrypt process memory pages
b) Extend RAM using disk space
c) Reduce network packet latency
d) Improve CPU clock stability

74. Which storage technology offers non-volatile memory with low latency and no mechanical parts?

a) Spinning hard disk
b) Solid-state drive flash
c) Magnetic tape reels
d) Optical Blu-ray disc

75. Which protocol is commonly used to establish reliable, ordered communication between applications over IP networks?

a) UDP datagram service
b) TCP transmission control
c) ARP address resolution
d) ICMP control messages

76. A hash function used in cryptography should primarily provide which property?

a) Deterministic collision resistance
b) Guaranteed compression expansion
c) Easy reversibility property
d) Predictable output variance

77. When optimizing a parallel program, which bottleneck is best addressed using Amdahl’s Law analysis?

a) Network jitter variability
b) GPU memory bandwidth
c) Serial fraction limitation
d) I/O interrupt latency

78. Which data structure offers average O(1) lookup time using key-based indexing?

a) Priority queue heap
b) Linked list nodes
c) Hash table mapping
d) Binary search tree

79. In relational databases, which operation combines rows from two tables based on a matching condition?

a) Aggregation group sum
b) Union set merge
c) Projection column subset
d) Join relational match

80. For secure web communication, which protocol layer provides end-to-end encryption between client and server?

a) TLS over HTTP connections
b) SSH for remote shells
c) DNS over UDP queries
d) SMTP with STARTTLS email

Answer Keys

1. c) Provide a convenient program execution environment
2. d) Distributes hardware resources fairly and efficiently
3. a) Supervision of program execution to prevent errors
4. a) It prioritizes user interaction over raw utilization
5. c) Maximize usability for the user
6. b) Improved interaction can enhance overall efficiency
7. c) Meeting fixed time constraints for tasks
8. a) Breakdown of the entire system behavior
9. d) Supervision of execution and I/O device management
10. a) Enabling rich GUI features to improve user experience
11. c) User mode lacks access to certain CPU instructions
12. b) Accessing an I/O device
13. b) Modify entries in device-status table
14. c) To prevent untrusted code from blocking critical events
15. d) Issue a trap instruction
16. a) It introduces security vulnerabilities and bloat
17. d) Restricted execution of certain instructions
18. d) Set timer, clear memory, access I/O
19. d) System could miss critical device events
20. b) Better access to kernel features for performance
21. a) Unauthorized access to OS data becomes possible
22. a) Provide finer-grained security among user groups
23. c) Users in the same group execute each other’s code
24. d) Run in a quasi-user/kernel mode for service
25. a) Set a future timer, sleep, update state on interrupt
26. b) Continually set interrupts and update local state
27. b) They buffer with intermediate speed between components
28. c) Bridging speed mismatch during data transfers
29. d) Maintain consistency with component data
30. d) Keeping cached data consistent with device
31. c) Clients request services that servers provide
32. c) As peers that may act as client or server
33. c) All recipes stored on a central server node

34. b) They may both request and provide recipes
35. d) The cache containing the datum must be updated
36. d) Cache and component have equal state-saving capacity
37. a) It ensures data persists without electricity
38. d) Faster storage tends to be more expensive
39. a) Provide a service to other peers
40. a) Strict roles separating client and server
41. b) Allow user processes to request OS services
42. a) Reads commands and invokes system calls
43. c) It changes frequently compared to kernel
44. a) fork() followed by exec()
45. d) Clones the current executing process
46. b) Replaces the process image with executable
47. d) Bundles of useful system calls
48. b) Basic functionality without writing code
49. c) Easier debugging and modification
50. a) Potential performance overhead across layers
51. c) Program execution
52. d) I/O operations
53. c) File manipulation
54. b) Communications
55. d) Hardware and software
56. e) They lack scheduling control
57. c) They might bypass protection
58. e) Deallocating file blocks
59. b) Device access control
60. b) Centralized error handling
61. a) Embedded device without a usable file system
62. d) Select which operating system to boot
63. d) Boot managers are designed with a default
64. c) At fixed locations on the hard disk
65. a) A boot manager presenting selections
66. c) Disk storage allows flexible file systems
67. c) Set the boot manager timeout and default
68. b) Boot manager program
69. c) Store the OS image within firmware
70. d) Operating systems stored at known locations
71. d) Kernel and scheduler core
72. b) Arithmetic Logic Unit
73. b) Extend RAM using disk space
74. b) Solid-state drive flash
75. b) TCP transmission control
76. a) Deterministic collision resistance
77. c) Serial fraction limitation
78. c) Hash table mapping

79. d) Join relational match

80. a) TLS over HTTP connections