4.5 Real-Time operating and Control System (ACtE0405)

Operating System Basics

1. Which of the following is not a function of an operating system?
    - a) Memory Management
    - b) Process Management
    - c) Web Browsing
    - d) File System Management

2. The operating system acts as an interface between:
    - a) User and software
    - b) Software and hardware
    - c) User and hardware
    - d) All of the above

3. Which component of the OS is responsible for managing memory allocation?
    - a) Kernel
    - b) CPU
    - c) Memory Manager
    - d) File System Manager

4. The core of an operating system is:
    - a) Shell
    - b) Command interpreter
    - c) Kernel
    - d) Application program

5. Which type of OS allows multiple users to interact with the computer at the same time?
    - a) Single-user OS
    - b) Multi-user OS
    - c) Batch OS
    - d) Real-time OS

6. Which of the following is a real-time operating system?
    - a) Linux
    - b) Windows
    - c) RTOS
    - d) Unix

7. The command line interface (CLI) is a type of:
    - a) GUI
    - b) Shell
    - c) Kernel
    - d) Memory manager

8. Which of the following is a function of the operating system?
    - a) Virus protection
    - b) Text editing
    - c) I/O Management
    - d) Data compression

Task, Process, and Threads

9. A process in execution refers to:
    - a) A task
    - b) A program
    - c) An application
    - d) An OS

10. Which of the following is true about processes?
    - a) Processes do not require memory.
    - b) Processes share memory by default.
    - c) Each process has its own memory space.
    - d) Processes are independent of each other.

11. A thread is also called a:
    - a) Process
    - b) Lightweight process
    - c) Heavyweight process
    - d) Program

12. Which of the following is shared among threads in the same process?
    - a) Stack
    - b) Registers
    - c) Program counter
    - d) Memory space

13. The unique identifier for a process is called its:
    - a) Process handle
    - b) Process ID (PID)
    - c) Thread ID
    - d) System ID

14. Which of the following is true about user-level threads?
    - a) They are managed by the kernel.
    - b) They are faster to create than kernel threads.
    - c) They cannot be created by the user.
    - d) They use more resources than kernel threads.

15. The state of a process is defined by:
    - a) Its memory usage
    - b) The activity of its threads
    - c) Its execution context (e.g., registers, program counter)
    - d) The number of open files

16. Which of the following is a benefit of using threads?
    - a) Simplifies memory management
    - b) Increases parallelism
    - c) Reduces the complexity of process scheduling
    - d) Improves single-threaded performance

17. Which of the following is a valid state for a process?
    - a) Running, Waiting, Terminated
    - b) New, Running, Sleeping
    - c) Active, Suspended, Waiting

- d) Initializing, Running, Waiting

18. When a process creates a new process, the new process is called:
    - a) Parent process
    - b) Child process
    - c) Thread
    - d) Clone

19. Which of the following is not true about kernel-level threads?
    - a) They are slower to create compared to user-level threads.
    - b) They are managed by the operating system kernel.
    - c) They cannot be preempted.
    - d) They can run on different processors in a multi-core system.

20. Context switching is:
    - a) Saving the state of one process and loading the state of another.
    - b) The process of switching from one OS to another.
    - c) Moving data from the CPU to memory.
    - d) A method of deadlock prevention.

Task Scheduling

21. Which scheduling algorithm assigns the CPU to the process that arrives first?
    - a) Round Robin
    - b) Priority Scheduling
    - c) Shortest Job Next
    - d) First-Come, First-Served

22. In which scheduling algorithm does each process get a fixed time slice?
    - a) Priority Scheduling
    - b) Round Robin
    - c) Multilevel Queue Scheduling
    - d) Shortest Job Next

23. Which scheduling algorithm selects the process with the shortest execution time first?
    - a) First-Come, First-Served
    - b) Shortest Job Next
    - c) Priority Scheduling
    - d) Round Robin

24. In priority scheduling, a process with higher priority:
    - a) Always gets more CPU time.
    - b) Is executed before processes with lower priority.
    - c) Gets less CPU time.
    - d) Waits for processes with lower priority to finish.

25. The issue of starvation in scheduling can occur when:
    - a) Processes are not given enough memory.
    - b) Lower-priority processes never get scheduled.
    - c) Processes get stuck in an infinite loop.
    - d) The system runs out of resources.

26. Which of the following scheduling algorithms can cause starvation?

- a) Round Robin
- b) First-Come, First-Served
- c) Priority Scheduling
- d) Shortest Job First

27. Which scheduling policy is most suitable for time-sharing systems?
    - a) First-Come, First-Served
    - b) Shortest Job First
    - c) Round Robin
    - d) Priority Scheduling

28. In a preemptive scheduling algorithm:
    - a) A running process cannot be interrupted.
    - b) A running process can be preempted by a higher-priority process.
    - c) All processes are executed in the order they arrive.
    - d) Processes are executed based on their burst time.

29. Which of the following is true for non-preemptive scheduling?
    - a) A process can be interrupted during execution.
    - b) A process holds the CPU until it completes or blocks.
    - c) Scheduling decisions are based on real-time constraints.
    - d) Higher-priority processes can interrupt running processes.

30. Which scheduling algorithm gives equal CPU time to all processes?
    - a) First-Come, First-Served
    - b) Shortest Job First
    - c) Round Robin
    - d) Priority Scheduling

31. Which of the following scheduling algorithms is non-preemptive?
    - a) Shortest Job First
    - b) Round Robin
    - c) Priority Scheduling
    - d) First-Come, First-Served

<u>Task Synchronization</u>

32. Task synchronization is essential in:
    - a) Single-threaded applications
    - b) Multithreaded applications
    - c) Operating systems only
    - d) User-level applications only

33. Which of the following is used to ensure mutual exclusion?
    - a) Semaphore
    - b) File system
    - c) CPU scheduling
    - d) Memory management

34. A critical section is:
    - a) A part of the process that performs I/O operations.
    - b) A segment of code that accesses shared resources.
    - c) A part of the operating system that handles scheduling.

- d) An error in process execution.

35. A deadlock is:
    - a) A situation where processes run in an infinite loop.
    - b) A situation where two or more processes wait indefinitely for resources held by each other.
    - c) A situation where the OS crashes due to memory overload.
    - d) A situation where multiple processes share the same resources.

36. Which of the following is not a condition for deadlock?
    - a) Mutual exclusion
    - b) Hold and wait
    - c) Preemption
    - d) Circular wait

37. Which mechanism can be used to avoid deadlock?
    - a) Process priority
    - b) Resource allocation graph
    - c) Scheduling algorithm
    - d) File system

38. A semaphore is a synchronization tool used to:
    - a) Control access to shared resources.
    - b) Allocate memory to processes.
    - c) Schedule CPU tasks.
    - d) Manage I/O operations.

39. Which of the following can lead to a race condition?
    - a) Improper synchronization
    - b) Deadlock prevention
    - c) Priority scheduling
    - d) Round Robin scheduling

40. Which method is commonly used to avoid race conditions?
    - a) Mutual exclusion
    - b) Preemption
    - c) Round Robin scheduling
    - d) Memory segmentation

41. In the context of synchronization, the term "busy waiting" refers to:
    - a) A process that waits for resources by continuously checking a condition.
    - b) A process that is ready to run but waiting for the CPU.
    - c) A thread that is waiting for a signal.
    - d) A method to allocate memory to processes.

42. Deadlock can be avoided by:
    - a) Implementing circular wait.
    - b) Allocating all required resources at the start.
    - c) Allowing preemption of resources.
    - d) Using non-preemptive scheduling.