Code: **Name:** **Class:**

1. **In Linux, what is the first process launched by the kernel after system boot, and what is its PID?**

a) systemd (with PID 1) – It initializes the runtime environment and manages other processes.

b) bash (with PID 0) – The default shell for user interactions.

c) gnome-terminal (with PID 2) – A graphical terminal emulator.

d) init (with PID 1) – An older alternative to systemd in legacy systems.

2. What is the fundamental difference between a program and a process in Linux?

a) A program is a static entity (source/executable code), while a process is dynamic with a changing state.
b) A program runs in the background, while a process runs in the foreground.
c) A program requires a PID, while a process does not.
d) A program is stored in RAM, while a process is stored on disk.

3. Which of the following processes is always assigned PID 1 in modern Linux systems?

a) gnome-session – Manages the user's desktop environment.
b) systemd – The init system responsible for boot and service management.
c) sshd – The Secure Shell daemon for remote access.
d) kthreadd – A kernel thread for internal scheduling.

4. How are new processes created in Linux?

a) Only by the kernel during system startup.
b) By an existing process (except systemd) through system calls like fork().
c) Directly by hardware interrupts.
d) Exclusively by the root user via terminal commands.

5. When a parent process creates a child process, what is this relationship called?

a) Parent-child hierarchy – The child inherits certain attributes from the parent.
b) Master-slave dependency – The child cannot execute independently.
c) Peer-to-peer linkage – Both processes share the same priority.
d) Client-server model – The child requests services from the parent.

6. Which of the following is NOT stored in a Process Control Block (PCB)?

a) Process state (e.g., running, waiting).
b) Process ID (PID) and parent PID.
c) The original source code of the program.
d) CPU register values (PC, SP, IR, etc.).

7. Which CPU register holds the memory address of the next instruction to be executed?

a) Program Counter (PC) – Directs the CPU to the next instruction.
b) Instruction Register (IR) – Stores the currently executing instruction.
c) Stack Pointer (SP) – Tracks the top of the runtime stack.
d) Status Flag (SF) – Indicates arithmetic/logic operation results.

8. What is the primary drawback of single-tasking operating systems?

a) CPU idleness during I/O operations – No other process can run while waiting.
b) Inability to run graphical applications.
c) Lack of process isolation.
d) High memory consumption.

9. In concurrent processing, which two queues are essential for managing process execution?
a) Job queue and interrupt queue.
b) Ready queue (for executable processes) and I/O wait queue (for blocked processes).
c) Memory queue and disk queue.
d) User queue and kernel queue.

10. When does a context switch occur in a multitasking system?
a) When a process requests a file operation.
b) When the CPU switches from executing one process to another.
c) When the system boots up.
d) When a child process is created using fork().

11. Which system call creates a child process with fine-grained control over shared resources?
a) fork() – Duplicates the parent process entirely.
b) clone() – Allows selective sharing of memory, file descriptors, etc.
c) exec() – Replaces the current process image.
d) vfork() – Suspends the parent until the child exits.

12. Which system call replaces the current process's executable code with a new program?
a) fork() – Creates a copy of the current process.
b) wait() – Pauses the parent until the child finishes.
c) exec() – Overwrites the process with a new executable.
d) vfork() – Temporarily suspends the parent.

13. A process running without user interaction is classified as:
a) A background process – Runs independently of the terminal.
b) A foreground process – Requires user input/output.
c) A zombie process – Has terminated but remains in the process table.
d) A daemon process – A specialized background service.

14. Which of the following is NOT a valid process state in Linux?
a) Running – Actively executing on the CPU.
b) Waiting – Blocked for I/O or an event.
c) Compiling – A transient phase during program creation (not a runtime state).
d) Ready – Prepared to execute but waiting for CPU time.

15. In the fork() system call, how do the parent and child processes differ?
a) They share the same PID.
b) They have distinct PIDs and independent memory spaces.
c) The child cannot execute concurrently with the parent.
d) The child inherits all open file descriptors but cannot modify them.

16. Which system call suspends the parent process until a child process terminates?
a) exec() – Replaces the process image.
b) wait() – Blocks the parent until the child exits.
c) clone() – Creates a customizable child process.
d) vfork() – Shares memory with the parent temporarily.

17. Which CPU register stores the results of the most recent arithmetic operation?
a) Program Counter (PC).
b) Instruction Register (IR).
c) Status Flag (SF) – Contains condition codes (e.g., zero, overflow).
d) Stack Pointer (SP).

18. What is the key advantage of concurrent processing over single-tasking?
a) Improved CPU utilization – Other processes run while one waits for I/O.
b) Simpler process scheduling.
c) Lower memory usage.
d) No need for process synchronization.

19. Which system call creates a child process while suspending the parent until the child exits?
a) fork() – Parent and child run concurrently.
b) vfork() – Parent pauses to avoid memory conflicts.
c) exec() – Replaces the current process.
d) clone() – Offers resource-sharing flexibility.

20. The "Process location" field in a PCB refers to:
a) The physical memory address of the process.
b) The queue where the process resides (e.g., ready queue, I/O wait queue).
c) The directory containing the executable file.
d) The CPU core executing the process.

21. Which of the following is an example of a foreground process?
a) A running gnome-terminal instance accepting user commands.
b) A cron job executing scheduled tasks.
c) The systemd init process.
d) A background find operation scanning files.

22. When a process requests I/O, where is it typically moved?
a) The ready queue – For immediate CPU access.
b) The I/O wait queue – Until the operation completes.
c) The job queue – For long-term scheduling.
d) The interrupt queue – For hardware signal handling.

23. A process in the "terminating" state is:
a) Waiting for user input.
b) Completing its execution and releasing resources.
c) Starting its initialization.
d) Paused due to a scheduler decision.

24. What is the primary role of systemd in Linux?
a) Compiling kernel modules.
b) Initializing the runtime environment and managing services/processes.
c) Directly controlling hardware devices.
d) Executing user-level applications.

25. After saving a process's state into its PCB during a context switch, what happens next?
a) The CPU loads another process's state from its PCB.
b) The terminated process is removed from memory.
c) The scheduler enters an idle loop.
d) The kernel checks for pending interrupts.

26. Which system call is NOT used to create a new process?
a) fork() – Standard process duplication.
b) vfork() – Lightweight creation with shared memory.
c) exec() – Replaces the current process (no new PID).
d) clone() – Customizable process creation.

27. What is the purpose of the Stack Pointer (SP) register?
a) To fetch the next instruction.
b) To track the top of the process's runtime stack (for function calls/local variables).
c) To store arithmetic results.
d) To hold interrupt masks.

28. In a single-tasking system, what happens when a process performs I/O?
a) Another process takes over the CPU.
b) The CPU remains idle until the I/O completes.
c) The operating system crashes.
d) The process is terminated automatically.

29. How can a parent process detect when its child terminates?
a) Using wait() – Blocks until the child exits.
b) Polling the child's PID.
c) Checking the /proc filesystem.
d) Receiving a kernel interrupt.

30. What resource is shared between parent and child when using vfork()?
a) The Process ID (PID).
b) The parent’s memory space (to avoid duplication).
c) The PCB.
d) CPU register values.