Operating Systems Revision Test - Week 6
Duration: 1 hour | **Total Questions:** 24

Good Luck!

SECTION A: MULTIPLE CHOICE QUESTIONS (15 Questions - 30 Marks)
Instructions: Choose the correct answer. Each question carries 2 marks.

1. Which of the following is NOT a primary function of an operating system?
- a) Process management
- b) Memory management
- c) Compiling source code
- d) File system management

2. In a layered architecture approach, which layer typically interacts directly with the hardware?
- a) Application layer
- b) System call interface
- c) Kernel layer
- d) User interface layer

3. Which of the following operating systems uses a microkernel architecture?
- a) Traditional Linux
- b) Windows NT (original design)
- c) MS-DOS
- d) Traditional Unix

4. What is the main advantage of a monolithic kernel over a microkernel?
- a) Better security
- b) Easier to maintain
- c) Better performance due to reduced context switching
- d) More modular design

5. Which component is responsible for loading the operating system kernel during the boot process?
- a) BIOS/UEFI
- b) Bootloader
- c) Init system
- d) Shell

6. What does GNU stand for in GNU/Linux?
- a) General Network Utilities
- b) GNU's Not Unix
- c) Global Network Unix
- d) General Nucleus Utilities

7. Which CPU scheduling algorithm can cause starvation of longer processes?
- a) First-Come-First-Serve (FCFS)
- b) Round Robin (RR)
- c) Shortest Job First (SJF)

- d) Priority Scheduling with aging

8. A race condition occurs when:

- a) Multiple processes compete for CPU time
- b) The outcome depends on the non-deterministic ordering of execution
- c) A process runs faster than expected
- d) The system runs out of memory

9. Which synchronization mechanism uses a counter to control access to shared resources?

- a) Mutex
- b) Monitor
- c) Semaphore
- d) Spinlock

10. In the context of memory management, which algorithm selects the smallest hole that is big enough for a process?

- a) First-Fit
- b) Best-Fit
- c) Worst-Fit
- d) Next-Fit

11. Which type of fragmentation occurs in paging?

- a) External fragmentation only
- b) Internal fragmentation only
- c) Both internal and external fragmentation
- d) No fragmentation occurs

12. What is the key difference between a pipe and a FIFO (named pipe)?

- a) Pipes are bidirectional, FIFOs are unidirectional
- b) FIFOs have a name in the filesystem and can be used by unrelated processes
- c) Pipes can work over networks, FIFOs cannot
- d) FIFOs are faster than pipes

13. Which signal is sent to a process when the user presses Ctrl+C in a Unix/Linux terminal?

- a) SIGKILL
- b) SIGTERM
- c) SIGINT
- d) SIGSTOP

SECTION B: NUMERICAL QUESTIONS - CPU SCHEDULING (5 Questions - 30 Marks)
Instructions: Show all calculations. Each question carries 6 marks.

Question 14: Consider the following processes with their arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

Calculate the average waiting time and average turnaround time using **First-Come-First-Serve (FCFS)** scheduling algorithm.

Hints:

1. **Draw Gantt Chart:** Execute processes in arrival order (P1→P2→P3→P4)
2. **Find Completion Time (CT):** When each process finishes
    - First process: CT = AT + BT
    - Others: CT = Previous CT + Current BT
3. **Calculate Turnaround Time (TAT):** TAT = CT - AT
4. **Calculate Waiting Time (WT):** WT = TAT - BT
5. **Find Averages:** Sum all WT / 4 and Sum all TAT / 4

Helpful Table:

Process	AT	BT	CT	TAT	WT
P1	0	8	?	?	?
P2	1	4	?	?	?
P3	2	9	?	?	?
P4	3	5	?	?	?

Question 15: Using the same process table from Question 16, calculate the average waiting time and average turnaround time using **Shortest Job First (SJF) Non-Preemptive** scheduling algorithm.

SECTION C: SCENARIO BASED QUESTIONS (10 Questions - 40 Marks)
Instructions: Analyze each scenario and provide detailed answers.

Question 16 (3 marks): A system administrator notices that a web server's response time has increased significantly. The CPU usage is at 25%, memory usage is at 40%, but disk I/O wait time is at 85%. Identify the bottleneck and suggest two solutions.

Question 17 (4 marks): A memory allocation request arrives for 150 KB. The available memory holes are: 100 KB, 250 KB, 180 KB, and 300 KB.

- (a) Which hole would be selected using **Best-Fit** algorithm?
- (b) Which hole would be selected using **Worst-Fit** algorithm?
- (c) What would be the remaining hole sizes after allocation in each case?
- (d) Which approach results in less external fragmentation here?

Question 18 (4 marks): In a multi-threaded banking application, two threads simultaneously try to withdraw money from the same account. Thread 1 tries to withdraw $500, and Thread 2 tries to withdraw $300. The current balance is $600.

- (a) Explain what problem could occur without proper synchronization.
- (b) Name the synchronization mechanism you would use to solve this problem.

Question 19 (5 marks): A system is running 5 CPU-bound processes and 5 I/O-bound processes. The current CPU utilization is 95%, but overall system throughput is low and I/O devices show 30% utilization.

- (a) Identify the possible bottleneck.
- (b) Suggest two specific improvements to optimize the system.

- (c) Which scheduling algorithm would be more suitable for this workload mix and why?

Question 20 (3 marks): You need to design IPC for a parent process to send configuration data to a child process that it spawns. The data is a one-time transfer of approximately 2 KB.

- (a) Which IPC mechanism would you choose from: pipe, FIFO, shared memory, or socket?
- (b) Justify your choice with two reasons.

Question 21 (5 marks): Compare the use of message queues versus shared memory for IPC in the following scenario: Two processes need to exchange large amounts of data (100 MB) frequently.

- (a) Which mechanism would be more efficient and why?
- (b) List two advantages of the chosen mechanism.
- (c) List one disadvantage of the chosen mechanism.

Question 22 (4 marks): A system uses segmentation with the following segment table:

Segment	Base Address	Limit
0	2000	800
1	4300	1200
2	1500	500

- (a) Will the logical address (Segment 1, Offset 1500) generate a segmentation fault? Explain why.
- (b) Calculate the physical address for logical address (Segment 0, Offset 400).

Question 23 (4 marks): Two processes need to communicate bidirectionally over a network. Process A is on Machine X and Process B is on Machine Y.

- (a) Which IPC mechanism must be used?
- (b) What type of socket would you recommend (TCP or UDP) for a file transfer application and why?

Question 24 (4 marks):

You are developing a web server application that spawns multiple worker processes to handle client requests. The main server process (parent) manages these worker processes (children). During a system shutdown, the server must stop accepting new connections, allow current requests to complete (up to 30 seconds), clean up resources properly, and force terminate any workers that don't finish within the timeout.

(a) The system administrator wants to gracefully shut down the web server. Which signal should be sent to the main server process? (1 mark)

- a) SIGINT - because it's the fastest way to stop the server
- b) SIGKILL - because it ensures the process stops immediately
- c) SIGTERM - because it can be caught and allows graceful shutdown
- d) SIGCHLD - because it notifies the parent to stop children

(b) After the parent receives the shutdown signal, what is the correct signal flow? (1 mark)

- a) Parent sends SIGTERM to children → Children exit → Kernel sends SIGCHLD to parent
- b) Parent sends SIGKILL to children → Children exit → Parent sends SIGTERM to kernel
- c) Parent sends SIGCHLD to children → Children send SIGTERM to parent
- d) Parent waits for SIGCHLD → Then sends SIGTERM to children

(c) A child worker process hasn't completed after the 30-second timeout. What should the parent do? (1 mark)

- a) Send another SIGTERM and wait longer
- b) Send SIGKILL to force immediate termination
- c) Send SIGINT to interrupt the child
- d) Ignore it and exit anyway

(d) A developer suggests the following code for the parent's signal handler:

```
void shutdown_handler(int sig) {
   for (each child) {
      kill(child_pid, SIGTERM);
   }
   exit(0);  // Exit immediately
}
```

What is the main problem with this implementation? (1 mark)

- a) Should use SIGKILL instead of SIGTERM for children
- b) Parent exits immediately without waiting for children to finish, creating zombie processes
- c) The signal handler should not call kill() function
- d) Should send SIGINT instead of SIGTERM
