OPERATING SYSTEM Paper 2

SECTION A

1. • Describe how the Von Neumann architecture differs from early computer systems.
2. • Explain the role of the control unit in the fetch-execute cycle.
3. • What are the key differences between the ALU and the CU in a CPU?
4. • How does bus width affect a computer's performance?
5. • Explain the function of the memory data register (MDR) in the fetch-execute cycle.
6. • How does a CPU use cache memory to enhance performance?
7. • Compare unidirectional and bidirectional buses.
8. • Explain how the accumulator register is used during arithmetic operations.
9. • Describe the difference between logical shifts and arithmetic shifts.
10. • Explain how bit manipulation is used for device control.
11. • What are the advantages of USB over older port technologies?
12. • How does increasing clock speed affect a CPU's performance?
13. • Explain the fetch-execute cycle using register transfer notation (RTN).
14. • What is the purpose of a status register in CPU operations?
15. • How does an interrupt service routine (ISR) handle an interrupt?
16. • Describe the function of the program counter (PC) during the fetch-execute cycle.
17. • How does a system clock synchronize data flow in a computer system?
18. • Explain the difference between synchronous and asynchronous data transmission.
19. • What is the role of the control bus in managing system operations?
20. • Compare and contrast the functions of the memory address register (MAR) and the memory data register (MDR).
21. • How does cache memory improve data retrieval speed?
22. • What are the limitations of VGA compared to HDMI in modern devices?
23. • How does an arithmetic shift differ from a logical shift in binary operations?
24. • What is meant by "masking" in bit manipulation?
25. • Explain how an interrupt handler works within the CPU.
26. • How does the fetch-execute cycle relate to the overall performance of the CPU?
27. • What is the difference between a flag and a register?

28. • Describe how overclocking affects CPU performance and stability.
29. • How do bitwise AND, OR, and XOR operations work in bit manipulation?
30. • Describe how the data bus interacts with memory and I/O devices.
31. • How does a CPU manage multiple cores during processing?
32. • Explain how binary shifts are used to multiply and divide by powers of two.
33. • How does HDMI prevent piracy using high-bandwidth digital copy protection (HDCP)?
34. • What is the role of the accumulator in assembly language programming?
35. • How does a CPU prioritize different interrupts?
36. • Describe how bus width and clock speed together influence system performance.
37. • What is the purpose of Register Transfer Notation (RTN) in describing CPU operations?
38. • How does a cyclic shift differ from a left or right logical shift?
39. • How do status flags help in error detection during CPU operations?
40. • Explain how an interrupt flag is set during the fetch-execute cycle.
41. • How is bit manipulation used in a control system with sensors?
42. • Explain the importance of cache memory in reducing CPU bottlenecks.
43. • How does the CPU handle interrupts while executing a task?
44. • Describe how address buses and data buses interact during the fetch-execute cycle.
45. • How does a CPU handle multiple interrupt requests?
46. • Explain the role of bitwise XOR in clearing bits in a register.
47. • How does dual-core technology improve system performance?
48. • What happens during a CPU's execute stage in the fetch-execute cycle?
49. • Explain how a CPU detects and services interrupts.
50. • How does a wider address bus improve memory access?

SECTION B

1. How does an operating system maximize CPU usage?
2. Explain the concept of virtual memory and its benefits.
3. Describe the differences between paging and segmentation.
4. What are the three areas of resource management in an OS?
5. How does the kernel manage device communication?
6. Compare preemptive and non-preemptive multitasking.
7. How does paging prevent data from running out of memory in an OS?
8. Explain how disk thrashing occurs and how it can be prevented.
9. Describe the process of context switching and its importance.
10. What is the difference between logical and physical memory?
11. How does the "shortest job first" (SJF) scheduling work?
12. How does the "shortest remaining time first" (SRTF) scheduling algorithm minimize waiting time?
13. What is the role of the process control block in memory management?
14. Describe the difference between dirty pages and clean pages in memory management.
15. What are the advantages and disadvantages of virtual memory?
16. Explain how DMA improves CPU efficiency.
17. Describe the stages in the process of handling an interrupt.
18. What are the key differences between paging and segmentation?
19. What is the role of the page table in memory management?
20. How does "Belady's Anomaly" affect page replacement algorithms?
21. Explain the differences between "round robin" and "first-come, first-served" scheduling.
22. Describe the importance of interrupt handling in multitasking.
23. Explain how the "least recently used" (LRU) page replacement algorithm functions.
24. What are the potential drawbacks of excessive use of virtual memory?
25. Describe the purpose of page replacement algorithms.
26. How do modern computers ensure the user-friendly transfer of data between devices?
27. How does preemptive multitasking ensure fairness between processes?
28. What are the roles of the three process states: running, ready, and blocked?
29. Explain how segmentation can lead to external fragmentation.
30. What is the purpose of process priorities in scheduling routines?

31. How does the kernel handle scheduling conflicts between processes?
32. How is the "clock" page replacement algorithm different from "LRU"?
33. What is a time slice, and how is it used in scheduling algorithms?
34. Explain the difference between internal and external fragmentation.
35. How does a system transition between different process states?
36. What is a swap file, and how does it function in virtual memory?
37. How does an OS handle process starvation?
38. Explain how memory is managed using segmentation.
39. How does page replacement ensure efficient memory use?
40. How does a non-preemptive system risk process starvation?
41. How does the DMA controller communicate with the CPU during data transfer?
42. What are the differences between static and dynamic memory allocation?
43. How does a system recover from a page fault?
44. Explain how process scheduling impacts CPU efficiency.
45. What is the role of interrupts in process management?
46. Describe how CPU-bound and I/O-bound processes differ in scheduling.
47. How does a command-line interface (CLI) provide greater control over hardware?
48. Explain the process of storing and retrieving data in a segmented memory system.
49. How does a system prevent two processes from using the same resource simultaneously?
50. Describe the relationship between burst time and process waiting time.