

Department : CSE(AI)

Year & Semester :III & V SEM

Sub Code & Sub Name :23CSE353T operating system and system programming

Unit-I

S.No	Part-A Questions
1.	Define an Operating System.
2.	List any two types of Operating Systems.
3.	What is a System Call?
4.	Define Virtual Machine.
5.	What is a Process Control Block (PCB)?
6.	List the different process states.
7.	Define Context Switching.
8.	What is a Thread?
9.	Mention two scheduling criteria.
10.	Define Multiprocessor Scheduling.
11.	What is a Race Condition?
12.	Define Critical Section.
13.	Write Peterson's Solution purpose.
14.	What is a Semaphore?
15.	Define Deadlock.

S.No	Part-B Questions
1.	Explain the generations and types of Operating Systems.
2.	Compare Layered, Monolithic, and Microkernel OS structures.
3.	Explain different OS services with examples.
4.	Describe the process of system boot.
5.	Write about process states with a neat diagram.
6.	Explain PCB and Context Switching in detail.
7.	Differentiate between Threads and Multithreading.
8.	Discuss CPU scheduling criteria with examples.
9.	Explain First-Come-First-Serve, SJF, and Round Robin scheduling.
10.	Describe multiprocessor scheduling and its performance evaluation.
11.	Explain process synchronization with Peterson's Solution.
12.	Discuss Semaphores and their applications.
13.	Explain Monitors with suitable examples.
14.	Write about Classical IPC Problems (Dining Philosophers/Readers-Writers).
15.	Discuss Deadlock prevention, avoidance, detection, and recovery.

Unit-II

S.No	Part-A Questions
1.	Differentiate between Logical and Physical Address.
2.	What is Internal Fragmentation?
3.	Define External Fragmentation.
4.	What is Compaction?
5.	Define Paging.
6.	What is a Page Fault?
7.	List any two Page Replacement Algorithms.
8.	Define Thrashing.
9.	What is Segmentation?
10.	Write the types of File Access Methods.
11.	What is a Directory Structure?
12.	Define File Allocation Methods.
13.	What is RAID?
14.	Define Disk Scheduling.
15.	What is Stable Storage?

S.No	Part-B Questions
1.	Explain contiguous memory allocation with advantages and disadvantages?
2.	Differentiate between paging and segmentation.
3.	Explain demand paging with a neat diagram.
4.	Describe page replacement algorithms (FIFO, LRU, Optimal).
5.	Explain thrashing and working set model.
6.	Discuss file system structure with neat diagram.
7.	Write about different directory structures.
8.	Explain file allocation methods with examples.
9.	Explain free space management techniques.
10.	Discuss disk structure and disk scheduling algorithms.
11.	Explain RAID levels with advantages.
12.	Describe swap space management.
13.	Explain tertiary storage with example.
14.	Compare logical vs physical address mapping.
15.	Discuss segmentation in detail with advantages.

Unit-III

S.No	Part-A Questions
1.	Define Device Controller.
2.	What is DMA (Direct Memory Access)?
3.	Write the role of an Interrupt Handler.
4.	Define Device Driver.
5.	What is Device-Independent I/O software?
6.	Define Protection Domain.
7.	What is User Authentication?
8.	List any two security design principles.
9.	Define Access Control List (ACL).
10.	What is the role of the Kernel?
11.	Write any two elementary Linux commands.
12.	What is Shell in Linux?
13.	Define System Call in Unix/Linux.
14.	What is System Administration?
15.	Define Directory Structure in Linux.

S.No	Part-B Questions
1.	Explain I/O hardware and device controllers.
2.	Describe DMA with a neat diagram.
3.	Explain I/O software layers with examples.
4.	Discuss device drivers and their role.
5.	Explain different security design principles.
6.	Discuss user authentication techniques.
7.	Explain protection mechanisms in an operating system.
8.	Write about Access Control Lists with examples.
9.	Explain the development of Unix/Linux.
10.	Discuss the role of the kernel in Unix/Linux.
11.	Explain system calls in Unix/Linux with examples.
12.	Write about shell programming basics.
13.	Explain elementary Linux commands with examples.
14.	Discuss Unix/Linux directory structure in detail.
15.	Explain responsibilities of a system administrator.

Unit-IV

S.No	Part-A Questions
1.	Define System Software.
2.	What is Software Hierarchy?
3.	Define Symbol Table.
4.	What is a Language Processor?
5.	Define Assembler.
6.	What is a Macro?
7.	Define Nested Macros.
8.	What is Linking?
9.	Define Loader.
10.	What is Relocation?
11.	Differentiate between Linker and Loader.
12.	What is Absolute Loader?
13.	Define Dynamic Linking.
14.	What is a Macro Assembler?
15.	Write any two data structures used in language processing.

S.No	Part-B Questions
1.	Explain system software and its hierarchy.
2.	Describe system programming and machine structure.
3.	Explain language processors with neat diagram.
4.	Write about symbol tables and their operations.
5.	Discuss search and allocation data structures in language processing.
6.	Explain design and types of assemblers.
7.	Describe macro definitions, expansion, and nested macros.
8.	Explain advanced macro features with examples.
9.	Write about macro assemblers and macro processors.
10.	Explain linkers and loaders with examples.
11.	Discuss relocation and its importance.
12.	Explain linking in MS-DOS with examples.
13.	Describe different loading schemes.
14.	Compare linkers and loaders.
15.	Discuss recent trends in system software development.

Unit-V

S.No	Part-A Questions
1.	Define Scanning.
2.	What is Parsing?
3.	Define Grammar Ambiguity.
4.	What is a Compiler?
5.	Define Interpreter.
6.	What is Semantic Gap?
7.	Define Binding.
8.	What are Scope Rules?
9.	What is Code Optimization?
10.	Define Debugger.
11.	What is Fork in Unix/Linux?
12.	Define Exec system call.
13.	What is a Pipe in OS?
14.	Define Signal in Unix/Linux.
15.	What is Threading?

S.No	Part-B Questions
1.	Explain programming language grammars with examples.
2.	Discuss ambiguity in grammar specification.
3.	Explain scanning and parsing techniques.
4.	Differentiate between compilers and interpreters.
5.	Explain the compilation process in detail.
6.	Discuss semantic gap, binding, and scope rules.
7.	Explain memory allocation strategies in compilers.
8.	Write about compilation of expressions and control structures.
9.	Explain code optimization techniques.
10.	Write about interpreters and debuggers.
11.	Explain C development tools.
12.	Discuss file I/O operations in system programming.
13.	Explain process creation and control (fork, exec).
14.	Describe pipes and signals in Unix/Linux.
15.	Explain basic threading with examples.