

Chapter 1 : Introduction 1-1 to 1-27

Syllabus : Introduction : Computer system overview, Architecture, Goals & Structures of O.S, Basic functions, Interaction of O.S. & hardware architecture, System calls, Batch, multiprogramming, Multitasking, time sharing, parallel, distributed and real-time OS.

1.1	Introduction to Operating System.....	1-1
1.1.1	Need of Operating System.....	1-1
1.1.2	Views of Operating system	1-1
1.2	Computer System Overview	1-2
1.2.1	Processors.....	1-3
1.2.2	Memory.....	1-3
1.2.3	Magnetic Disks	1-4
1.2.4	I/O Devices	1-5
1.3	Computer System Architecture	1-7
1.3.1	Single-Processor Systems.....	1-7
1.3.2	Multiprocessor Systems.....	1-7
1.3.3	Cluster Systems.....	1-8
1.4	Goals & Structures of OS.....	1-8
1.4.1	Design Goals	1-8
1.4.2	Structures of OS	1-9
1.5	Operating System Objectives and Functions	1-9
1.5.1	Objectives of Operating System.....	1-9
1.5.2	Functions of Operating System.....	1-10
1.6	Interaction of O.S. and Hardware Architecture	1-11
1.6.1	Dual Mode and Multi Mode Operation	1-11
1.6.2	Privileged Operations.....	1-12
1.6.3	Timer	1-12
1.6.4	Memory Protection.....	1-12
1.7	System Calls.....	1-13
1.7.1	Types of System Calls	1-14
1.7.2	Some Examples of System Calls.....	1-15
1.8	Batch Operating System	1-15
1.9	Multiprogramming Operating System.....	1-15
1.10	Timesharing Operating System.....	1-16
1.11	Multitasking OS.....	1-16
1.12	Parallel Operating System	1-16
1.13	Distributed Operating System	1-17

1.13.1	Definition of Distributed Operating System.....	1-17
1.13.2	Motivation	1-17
1.13.3	Types of Distributed Operating Systems.....	1-18
1.13.3(A)	Network Operating System (NOS)	1-19
1.13.3(B)	Distributed Operating Systems (DOS)	1-19
1.13.4	Difference between Distributed OS and Network OS	1-20
1.14	Real Time Operating System	1-21
1.14.1	Characteristics of Real-Time Operating Systems (RTOSs)	1-21
1.15	Operating System Structure.....	1-23
1.15.1	Monolithic Systems.....	1-23
1.15.2	Layered Systems	1-24
1.15.3	Virtual Machines	1-25
1.15.4	Client-Server Model	1-26
1.15.5	Monolithic Kernel vs. Microkernel.....	1-26

Chapter 2 : Process and Threads Management

2-1 to 2-41

Syllabus : Process Concept, Process states, Process control, Threads, Uni-processor Scheduling : Types of scheduling : Preemptive, Non preemptive, Scheduling algorithms: FCFS, SJF, RR, Priority, Thread Scheduling, Real Time Scheduling. System calls like ps, fork, join, exec family, wait.

2.1	Process	2-1
2.1.1	Introduction to Process	2-1
2.1.2	Concept of Process	2-1
2.1.3	Context Switch	2-2
2.1.4	Process States	2-2
2.1.5	Process Description.....	2-3
2.1.5(A)	Operating System Control Structures.....	2-3
2.1.5(B)	Process Control Structures	2-3
2.1.6	Process Control Block	2-4
2.1.7	Process Control	2-5
2.1.7(A)	Execution Modes	2-5
2.1.7(B)	Process Creation	2-5
2.1.7(C)	Process Switching	2-6
2.1.7(D)	Change of Process State (Process Switch).....	2-7

2.1.7(E)	Loading Programs into Processes	2-7	2.5.4	Real Time Scheduling.....	2-32
2.1.7(F)	Process Termination.....	2-7	2.5.4(A)	Table Driven Scheduling.....	2-32
2.1.8	Execution of the Operating System.....	2-8	2.5.4(B)	Cyclic Schedulers	2-33
2.1.8(A)	No Process Kernel.....	2-8	2.5.4(C)	Earliest Deadline First (EDF) Scheduling :	2-35
2.1.8(B)	Execution within User Process.....	2-8	2.5.4(D)	Rate Monotonic Algorithm (RMA).....	2-37
2.1.8(C)	Process-Based Operating System	2-8	2.6	System Calls like ps, Fork, Join, exec Family, Wait.....	2-37
2.2	Threads	2-9	<hr/>		
2.2.1	Definition of Thread.....	2-9	Chapter 3 : Concurrency 3-1 to 3-8		
2.2.2	Processes and Threads	2-9	Syllabus : Concurrency : Principles of Concurrency, Mutual Exclusion: S/W approaches, H/W Support, Semaphores, Pipes, Message Passing, Signals, and Monitors.		
2.2.3	Concept of Multithreading	2-10	3.1	Principles of Concurrency.....	3-1
2.2.3(A)	Multithreading Models	2-10	3.2	Critical Section Problem.....	3-2
2.2.4	Types of Threads.....	2-11	3.3	Mutual Exclusion.....	3-4
2.2.4(A)	User Level Threads.....	2-11	3.3.1	Requirements for Mutual Exclusion.....	3-4
2.2.4(B)	Kernel Level Threads.....	2-12	3.3.2	Mutual Exclusion Conditions	3-4
2.2.4(C)	Hybrid Implementations	2-12	3.4	Hardware Support.....	3-4
2.2.5	Thread States	2-12	3.5	Software Approaches.....	3-5
2.3	Scheduling.....	2-12	3.5.1	Semaphores and Mutex.....	3-5
2.3.1	Scheduling Queues and Schedulers	2-12	3.5.2	Programming Language Support : (Monitors)	3-6
2.3.1(A)	Long-term Scheduler	2-13	3.5.3	Pipes	3-7
2.3.1(B)	Short-term Scheduler.....	2-13	3.6	Message Passing	3-7
2.3.1(C)	Medium-term Scheduler.....	2-13	3.7	Signals.....	3-8
2.3.1(D)	Comparison of Three Schedulers.....	2-14	<hr/>		
2.3.2	Types of Scheduling	2-14	Chapter 4 : Inter Process Communication 4-1 to 4-15		
2.3.3	Scheduling Algorithms	2-14	Syllabus : Race Conditions, Critical Section, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing. Classical IPC Problems: Reader's & Writer Problem, Dining Philosopher Problem etc., Scheduling, Scheduling Algorithms.		
2.3.3(A)	Scheduling Criteria.....	2-14	4.1	Introduction to Inter Process Communication	4-1
2.3.3(B)	Scheduling Algorithms	2-15	4.2	Race Conditions	4-2
2.3.3(C)	Examples on Scheduling Algorithms.....	2-20	4.3	Critical Section.....	4-3
2.4	Thread Scheduling.....	2-29	4.4	Mutual Exclusion.....	4-4
2.4.1	Advantages of Load Sharing.....	2-29	4.4.1	Requirement of Mutual Exclusion.....	4-4
2.4.2	Load Sharing Approaches.....	2-29	4.4.2	Mutual Exclusion Conditions	4-4
2.4.3	Gang Scheduling	2-30	4.4.3	Mutual Exclusion with Busy Waiting.....	4-5
2.4.4	Dedicated Processor Assignment	2-30	<hr/>		
2.4.5	Dynamic Scheduling	2-30			
2.5	Real Time Scheduling.....	2-31			
2.5.1	Types of Real Time Tasks	2-31			
2.5.2	Non Real Time Tasks	2-31			
2.5.3	Task Scheduling	2-32			

4.4.3(A) Disabling Interrupts4-5

4.4.3(B) Strict Alternation4-5

4.5 Hardware Solution (TSL Instruction)4-6

4.6 Peterson’s Solution.....4-6

4.7 Producer Consumer Problem.....4-7

4.8 Semaphores4-8

4.9 Producer Consumer Problem Using Semaphore4-9

4.10 Event Counters4-10

4.11 Monitors.....4-10

4.12 Message Passing4-12

4.13 Classical IPC Problems4-12

4.13.1 Reader’s & Writer Problem4-12

4.13.2 Dining Philosopher Problem.....4-13

Chapter 5 : Deadlock 5-1 to 5-12

Syllabus : Deadlock : Principles of Deadlock, Starvation, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, System calls.

5.1 Principles of Deadlock and Starvation.....5-1

5.1.1 System Model.....5-1

5.1.2 Resources5-1

5.1.3 Resource Allocation Graphs5-1

5.1.4 Conditions for Resource Deadlocks5-2

5.2 Deadlock Prevention.....5-3

5.2.1 Attacking the Mutual-Exclusion Condition5-3

5.2.2 Attacking the Hold-and-Wait Condition5-3

5.2.3 Attacking the No-Preemption Condition5-3

5.2.4 Attacking the Circular Wait Condition.....5-4

5.3 Deadlock Avoidance5-4

5.3.1 Safe and Unsafe States5-4

5.3.2 Deadlock Avoidance Algorithms5-5

5.3.2(A) Resource-Allocation Graph Algorithm5-5

5.3.2(B) Banker’s Algorithm.....5-6

5.3.2(C) Resource-Request Algorithm5-6

5.3.2(D) Safety Algorithm5-6

5.3.3 Solved Problems.....5-7

5.4 Deadlock Detection and Recovery5-9

5.4.1 Deadlock Detection with One Resource of Each Type5-9

5.4.2 Deadlock Detection with Multiple Resources of Each Type.....5-10

5.4.3 Recovery from Deadlock.....5-10

5.4.3(A) Process Termination (Kill a Process)5-11

5.4.3(B) Resource Preemption5-11

5.5 System Calls.....5-11

Chapter 6 : Memory Management 6-1 to 6-36

Syllabus : Memory Management : Requirements, Memory partitioning Fixed and Variable Partitioning, Memory Allocation: Allocation Strategies (First Fit, Best Fit, and Worst Fit), Swapping, Paging and Fragmentation. Demand Paging, Security Issues.

Virtual Memory : Concepts, VM management, Page Replacement Policies (FIFO, LRU, Optimal, Other Strategies), Thrashing.

6.1 Introduction to Memory Management.....6-1

6.1.1 No Memory Abstraction6-1

6.2 Memory Management Requirements6-2

6.2.1 Relocation.....6-2

6.2.2 Protection, Logical and Physical Address Space6-2

6.2.3 Sharing6-3

6.2.4 Logical Organization6-3

6.2.5 Physical Organization6-3

6.3 Memory Partitioning.....6-3

6.3.1 Mono-programming.....6-3

6.3.2 Multiprogramming6-4

6.3.3 Multiprogramming with Fixed and Variable Partitions (Contiguous Allocation)6-4

6.3.4 Dynamic Partition Technique6-5

6.3.5 Compaction6-6

6.4 Dynamic Loading.....6-6

6.5 Overlays6-6

6.6 Memory Allocation Strategies6-7

6.7 Buddy System6-8

6.8 Swapping6-9

6.9 Relocation.....6-9

6.10 Paging and Fragmentation.....6-10

6.10.1 Basic Operation6-10

6.10.2 Memory Protection and Sharing.....6-11

6.10.3 Translation Look aside Buffer6-12

6.10.4 Effect of Page Size on Performance6-12

6.10.5 Hardware Support for Paging.....6-12

6.11 Segmentation.....6-13

6.12 Segmentation with Paging6-15

6.13 Demand Paging6-16

6.14 Security Issues6-16

6.14.1 Buffer Overflow Attacks6-17

6.14.2 Defending Against Buffer Overflows.....6-17

6.15 Virtual Memory Management6-17

6.15.1 Virtual Memory.....6-17

6.16 Structure of Page Tables6-17

6.16.1 Hierarchical Paging.....6-17

6.16.2 Hashed Page Table6-19

6.16.3 Inverted Page Table.....6-19

6.17 Copy-on-Write.....6-20

6.18 Page Replacement Policies6-21

6.18.1 FIFO Algorithm6-21

6.18.2 Optimal Page Replacement Algorithm6-22

6.18.3 Least Recently Used Page Replacement Algorithm (LRU).....6-22

6.18.4 Implementation of LRU6-23

6.18.5 LRU-Approximation Page Replacement.....6-23

6.18.5(A) Additional-Reference-Bits Algorithm6-23

6.18.5(B) Second-Chance Algorithm6-24

6.18.5(C) Enhanced Second-Chance Algorithm6-24

6.18.5(D) Clock Page Replacement Algorithm.....6-24

6.18.6 Counting-Based Page Replacement6-25

6.18.6(A) Not Frequently Used or Least Frequently Used Page Replacement Algorithm (NFU or LFU)6-25

6.18.6(B) Most Frequently Used Page Replacement Algorithm (MFU)6-25

6.19 Examples on Page Replacement Policies.....6-25

6.20 Allocation of Frames6-34

6.21 Thrashing.....6-35

6.22 Locality (Working Set Model)6-35

Chapter 7 : I/O Management and Disk Scheduling

7-1 to 7-22

Syllabus : I/O Devices, Organization of I/O functions, Operating System Design issues, I/O Buffering, Disk Scheduling (FCFS, SCAN, C-SCAN, SSTF), RAID, Disk Cache.

7.1 I/O Devices7-1

7.2 Organization of the I/O Function7-2

7.2.1 Evolution of the I/O Function.....7-2

7.2.2 Direct Memory Access.....7-3

7.3 Operating System Design Issues.....7-4

7.3.1 Design Objectives.....7-4

7.3.2 Logical Structure of the I/O Function.....7-4

7.4 I/O Buffering7-6

7.4.1 Single Buffer7-7

7.4.2 Double Buffer.....7-8

7.4.3 Circular Buffer.....7-8

7.4.4 Utility of Buffering.....7-8

7.5 Disk Scheduling Algorithms7-8

7.5.1 FCFS Scheduling Algorithm.....7-9

7.5.2 Shortest-Seek-Time-First (SSTF) Scheduling Algorithm7-9

7.5.3 SCAN Scheduling Algorithm7-10

7.5.4 C-SCAN Scheduling Algorithm7-11

7.5.5 LOOK Scheduling Algorithm7-11

7.6 Examples on Disk Scheduling Algorithms.....7-12

7.7 RAID (Redundant Array of Inexpensive Disks).....7-18

7.8 Disk Cache7-21

Chapter 8 : Security and Protection

8-1 to 8-8

Syllabus : Security Environment, Design Principles of Security, User Authentication, Protection Mechanism: Protection Domain, Access Control List.

8.1 Security Environment.....8-1

8.1.1 Threats8-1

8.1.2 Trojan Horse and Trap Doors Program Threats8-2

8.1.3 Intruders8-2

8.1.4 Accidental Data Loss8-2

8.2 Design Principles of Security8-3

8.3	User Authentication.....	8-3	9.4.2(I)	File Tests.....	9-13
8.3.1	Passwords	8-3	9.4.2(J)	Security Issues in Shell Scripts.....	9-14
8.3.2	Physical Identification	8-4	9.4.2(K)	Shell Commands	9-14
8.3.3	Countermeasures	8-5	9.4.2(L)	Shell Script	9-14
8.4	Protection Mechanism	8-5	9.4.2(M)	Shell Script Examples.....	9-15
8.4.1	Protection Domains	8-5	9.5	Directory Structure.....	9-23
8.4.2	Access Control Lists	8-7	9.5.1	UNIX File Allocation	9-23
<hr/>					
Chapter 9 : Unix/Linux Operating System 9-1 to 9-93					
<hr/>					
Syllabus : Development Of Unix/Linux, Role & Function Of Kernel, System Calls, Elementary Linux command & Shell Programming, Directory Structure, System Administration Case study: Linux, Windows Operating System.					
9.1	Development of Unix/Linux	9-1	9.5.2	Directories	9-24
9.1.1	Introduction to UNIX.....	9-1	9.5.3	Mounting/Unmounting File Systems.....	9-25
9.1.2	History of UNIX.....	9-1	9.5.4	Important UNIX Directories and Files.....	9-25
9.1.3	History of Linux.....	9-3	9.6	System Administration.....	9-26
9.2	Role & Function of Kernel	9-4	9.7	Case Study: Linux Operating System	9-26
9.3	System Calls.....	9-4	9.7.1	Linux Design Principles and Goals.....	9-26
9.3.1	UNIX System Calls	9-5	9.7.1(A)	Linux Goals.....	9-26
9.3.1(A)	File System Calls in UNIX.....	9-5	9.7.1(B)	Utility Programs in Linux	9-27
9.3.1(B)	Directory System Calls in UNIX.....	9-5	9.7.2	Components of a Linux System	9-28
9.3.1(C)	Process Management System Calls in UNIX	9-6	9.7.3	Linux Booting.....	9-29
9.3.1(D)	Thread Management System Calls in UNIX.....	9-7	9.7.4	Kernel Modules.....	9-29
9.3.1(E)	Memory Management System Calls in UNIX	9-7	9.7.4(A)	Module Management	9-30
9.3.1(F)	I/O System Calls in UNIX.....	9-7	9.7.4(B)	Driver Registration	9-30
9.4	Elementary Linux commands & Shell Programming	9-8	9.7.4(C)	Conflict Resolution.....	9-31
9.4.1	Basic Linux Commands	9-8	9.7.5	Process Management.....	9-31
9.4.2	Shell Programming	9-10	9.7.5(A)	fork() and exec() Process Model	9-31
9.4.2(A)	Shell Scripts.....	9-11	9.7.5(B)	Processes and Threads	9-35
9.4.2(B)	History of shells	9-11	9.7.6	Linux Scheduling.....	9-35
9.4.2(C)	Features of Shell Programming	9-11	9.7.6(A)	Process Scheduling	9-35
9.4.2(D)	Features of the Shell Environment.....	9-11	9.7.6(B)	Kernel Synchronization	9-37
9.4.2(E)	Advantages of bash	9-11	9.7.6(C)	Real Time Scheduling.....	9-40
9.4.2(F)	Special Characters in Shell	9-12	9.7.6(D)	Symmetric Multiprocessing.....	9-41
9.4.2(G)	Execution of Commands.....	9-13	9.7.6(E)	Process Management System Calls in Linux	9-41
9.4.2(H)	Input/Output Redirection	9-13	9.7.7	Memory Management.....	9-42
			9.7.7(A)	Linux Virtual Memory	9-42
			9.7.7(B)	Operation of Buddy Algorithm	9-43
			9.7.7(C)	Linux Policy for Page Replacement	9-43
			9.7.7(D)	Simple Clock Algorithm for Page Replacement.....	9-44
			9.7.7(E)	Kernel Memory Allocation	9-44
			9.7.7(F)	Memory Management System Calls in Linux	9-45

9.7.8	Linux File Systems.....	9-46	9.8.5	Memory Management.....	9-81
9.7.8(A)	Virtual File System (VFS).....	9-46	9.8.5(A)	Fundamental Concepts.....	9-81
9.7.8(B)	Linux Ext2 File System	9-47	9.8.5(B)	Memory-Management System Calls	9-82
9.7.8(C)	Journaling	9-47	9.8.5(C)	Implementation of Memory Management.....	9-82
9.7.8(D)	Linux Process (/proc) File System.....	9-48	9.8.6	Caching in Windows	9-86
9.7.8(E)	NFS : Network File System	9-49	9.8.7	Input/Output in Windows.....	9-87
9.7.8(F)	File System Calls in Linux.....	9-50	9.8.8	Windows NT File System.....	9-88
9.7.8(G)	Directory System Calls in Linux.....	9-51	9.9	Windows Power Management	9-90
9.7.9	Linux Input Output.....	9-51	9.10	Security in Windows 8	9-90
9.7.9(A)	Networking.....	9-51	<hr/>		
9.7.9(B)	I/O System Calls	9-53	Chapter 10 : Virtualization Concepts 10-1 to 10-9		
9.7.9(C)	Implementation of Input/Output in Linux.....	9-53	Syllabus : Virtual machines ; supporting multiple operating systems simultaneously on a single hardware platform; running one operating system on top of another. True or pure virtualization.		
9.7.10	Linux Inter-Process Communication	9-56	10.1	History	10-1
9.7.10(A)	Synchronization and Signals.....	9-56	10.2	Benefits	10-3
9.7.10(B)	Passing of Data among Processes	9-57	10.3	Requirements for Virtualization	10-3
9.7.11	Linux Security	9-57	10.4	Supporting Multiple Operating Systems Simultaneously on a Single Hardware Platform	10-4
9.8	Case study : Windows Operating System	9-59	10.5	Running One Operating System on Top of Another	10-4
9.8.1	History of Windows	9-59	10.6	Techniques for Efficient Virtualization	10-6
9.8.2	Programming Windows.....	9-60	10.6.1	Virtualizing the Unvirtualizable	10-6
9.8.3	System Structure	9-65	10.6.2	Cost of Virtualization.....	10-8
9.8.4	Processes and Threads in Windows	9-75	10.7	True or Pure Virtualization	10-8
9.8.4(A)	Fundamental Concepts.....	9-75			
9.8.4(B)	Jobs and Fibers	9-76			
9.8.4(C)	Thread Pools and User-Mode Scheduling (UMS)....	9-76			
9.8.4(D)	Interprocess Communication	9-78			
9.8.4(E)	Implementation of Processes and Threads.....	9-79			