

**UNIT - I****Chapter 1 : Introduction 1-1 to 1-51**

1.1	System Programming.....	1-1
1.2	Software Hierarchy.....	1-1
1.3	Software Tools for Program Development.....	1-2
1.3.1	Editor.....	1-2
1.3.1(A)	Editor Structure.....	1-3
1.3.2	Debug Monitor.....	1-3
1.3.3	Programming Environment.....	1-4
1.3.4	User Interfaces.....	1-4
1.4	Basic Functions of Loaders.....	1-4
1.5	Assembly Language Programming.....	1-6
1.6	Macro Processor.....	1-6
1.7	Compiler.....	1-7
1.7.1	An Interpreter.....	1-8
1.7.2	Difference between Interpreter and Compiler.....	1-8
1.7.3	A Cross Compiler.....	1-8
1.7.4	A Bootstrap Compiler.....	1-9
1.8	The Von Neumann Architecture.....	1-9
1.8.1	Input Unit.....	1-9
1.8.2	Output Unit.....	1-10
1.8.3	Arithmetic and Logic Unit (ALU).....	1-10
1.8.4	Control Unit.....	1-10
1.8.5	Memory Unit.....	1-10
1.8.6	Key Features of a Von Neumann Machine.....	1-10
1.8.7	Basic Structure of the CPU.....	1-10
1.8.8	Execution of a Program by Von Neumann Machine.....	1-11
1.9	Assembly Language Programming.....	1-14

1.9.1	Elements of Assembly Language.....	1-14
1.9.2	Assembly Program to Machine Language Program.....	1-16
1.9.3	Assembly Language Statements.....	1-18
1.9.4	Literals and Constants.....	1-19
1.10	Assembler and Related Programs.....	1-20
1.10.1	Load and Go Assembler.....	1-20
1.10.2	One-pass Assembler.....	1-21
1.10.2(A)	One-pass Assembler with Forward References.....	1-21
1.10.2(B)	Design of Single Pass Assembler.....	1-22
1.10.3	Two Pass Assembler.....	1-22
1.11	Advanced Assembler Directives.....	1-23
1.12	Design of Two Pass Assembler.....	1-25
1.12.1	Intermediate Code.....	1-28
1.12.1(A)	Variants of Intermediate Code.....	1-29
1.12.2	Algorithm for Pass I.....	1-31
1.12.3	Flowchart for Pass I.....	1-34
1.12.4	C-Program for Pass I.....	1-36
1.12.5	Pass II of the Assembler.....	1-41
1.12.6	C-Program for Pass II.....	1-42
1.12.7	Error Reporting.....	1-47
1.13	Study of Debugging Tools Like GDB.....	1-48
1.14	University Questions and Answers.....	1-49

**UNIT - II****Chapter 2 : Macro Processor and Compilers****2-1 to 2-50**

2.1	Macro Definition.....	2-1
2.2	Defining a Macro.....	2-2



2.2.1	Calling a Macro .....	2-3	2.9.4	Intermediate Code Generation .....	2-34
2.2.2	Macro Expansion .....	2-3	2.9.5	Code Optimization .....	2-36
2.2.3	Macro with Keyword Parameters .....	2-4	2.9.6	Code Generation .....	2-36
2.2.4	Macro with Mixed Parameters.....	2-5	2.10	Intermediate Code for Array .....	2-40
2.2.5	Other Uses of Parameters.....	2-5	2.11	Compilation of Control Structure .....	2-40
2.3	Nested Macro Calls .....	2-6	2.11.1	if Statement .....	2-41
2.3.1	Nested Macro Definition .....	2-7	2.11.2	while Statement .....	2-42
2.4	Advanced Macro Facilities .....	2-7	2.11.3	Function and Procedure Call .....	2-42
2.4.1	Expansion Time Variables (EV).....	2-8	2.12	Code Optimization .....	2-42
2.5	Design of Macro Processor .....	2-9	2.12.1	Machine Dependent Optimization .....	2-45
2.5.1	Issues Related to the Design of a Simple Macro Preprocessor .....	2-9	2.13	University Questions and Answers .....	2-48
2.5.2	Databases used in Pass-1 of 2 Pass Macro Processor .....	2-11	<b>UNIT - III</b>		
2.5.3	Flow Chart for Pass 1.....	2-11	<hr/>		
2.5.4	Databases used in Pass-2 of 2 Pass Macro Processor .....	2-11	<b>Chapter 3 : Linkers and Loaders                      3-1 to 3-28</b>		
2.5.5	Flow Chart for Pass-2 .....	2-12	<hr/>		
2.5.6	A Simple One-Pass Macro Processor .....	2-12	3.1	Basic Functions of a Loader .....	3-1
2.5.7	C-Program for One-pass Macro Processor.....	2-13	3.2	Loading Schemes .....	3-3
2.6	Handling of Nested Macro Calls.....	2-18	3.2.1	Compile-and-go Loaders .....	3-3
2.7	Handling of Nested Macro Declaration.....	2-20	3.2.2	General Loader Scheme .....	3-3
2.8	Compiler.....	2-23	3.2.3	Absolute Loader.....	3-4
2.8.1	An Interpreter .....	2-24	3.2.4	Subroutine Linkage .....	3-5
2.8.2	Difference between Interpreter and Compiler ..	2-25	3.2.5	Relocating Loader .....	3-5
2.8.3	A Cross Compiler .....	2-25	3.2.6	Direct Linking Loader .....	3-7
2.8.4	A Bootstrap Compiler .....	2-25	3.3	Design of an Absolute Loader .....	3-9
2.9	General Model of Compiler .....	2-26	3.4	Design of Direct Linking Loader .....	3-10
2.9.1	Lexical Analysis .....	2-26	3.4.1	Format of Databases of a Direct Linking Loader .....	3-14
2.9.2	Syntax Analysis.....	2-33	3.4.2	Flowchart for Pass I .....	3-14
2.9.3	Semantic Analysis.....	2-33	3.4.3	Flowchart for Pass II.....	3-14
			3.5	Implementation of MS DOS Linker .....	3-15
			3.5.1	Data Structures and Algorithm.....	3-16



3.6	Overlay Structure .....	3-17	Interface.....	4-2	
3.7	Software Tools for Program Development.....	3-18	4.1.1(B) The Operating System as Resource Manager.....	4-3	
3.7.1	Editor.....	3-18	4.2	Evolution of Operating Systems.....	4-4
3.7.1(a)	Editor Structure.....	3-19	4.2.1	Serial Processing .....	4-4
3.7.2	Debug Monitor.....	3-19	4.2.2	Batch Processing.....	4-4
3.7.3	Programming Environment .....	3-20	4.2.3	Multiprogramming .....	4-5
3.7.4	User Interfaces.....	3-20	4.3	Types of Operating System.....	4-7
3.8	Relocation and Linking Concept .....	3-20	4.3.1	Batch Operating System.....	4-7
3.8.1	Linking.....	3-20	4.3.2	Multiprogramming Operating System .....	4-7
3.8.2	Relocation .....	3-21	4.3.3	Real-time System.....	4-8
3.8.3	Object Module .....	3-21	4.3.4	Network Operating System .....	4-8
3.9	Self Relocating Programs.....	3-22	4.3.5	Distributed Operating System.....	4-9
3.10	Static and Dynamic Link Libraries.....	3-22	4.4	Operating System Components .....	4-9
3.10.1	Static Linking.....	3-22	4.4.1	Processes.....	4-9
3.10.2	Dynamic Linking.....	3-22	4.4.2	Files .....	4-9
3.10.3	Dynamic Link Library (DLL).....	3-22	4.4.3	System Calls .....	4-10
3.10.4	Call Back Functions.....	3-23	4.4.4	Command Interpreter .....	4-11
3.10.5	Difference between Callback Function and Normal Function .....	3-24	4.4.5	Signals.....	4-11
3.11	Dynamic Linking with and without Import.....	3-24	4.5	Operating System Structure.....	4-12
3.11.1	Using the DLL (with an import library).....	3-25	4.5.1	Monolithic System.....	4-13
3.11.2	Using DLL (without an Important Library) .....	3-25	4.5.2	Layered Systems.....	4-13
3.12	Loading Phases using Java .....	3-26	4.5.3	Virtual Machine .....	4-13
3.13	University Questions and Answers.....	3-26	4.5.4	Client-Server Model .....	4-14
<b>UNIT - IV</b>			4.6	Process .....	4-14
<b>Chapter 4 : Operating System (OS) 4-1 to 4-54</b>			4.7	Process Control .....	4-15
4.1	Evolution of O.S. Functions .....	4-1	4.7.1	Process States (5 State Model) .....	4-15
4.1.1	Functions of an Operating System .....	4-2	4.7.2	Process States (7 State Model) .....	4-16
4.1.1(A)	The Operating System as a User/Computer		4.8	Process Implementation (Process Control Block) .....	4-16
			4.9	Threads .....	4-17



4.9.1	Benefits of Threads.....	4-18	4.18	Monitors.....	4-48
4.10	Process Scheduling .....	4-18	4.18.1	Bounded Buffer Producer-Consumer Problem with Monitor .....	4-49
4.10.1	Types of Schedulers .....	4-19	4.19	Pipes .....	4-50
4.11	Scheduling Methods .....	4-20	4.20	System Calls .....	4-51
4.11.1	First-Come-First-Served (FCFS) Scheduling .....	4-21	4.20.1	System Call Signal .....	4-51
4.11.1(A)	FCFS Algorithm.....	4-22	4.20.2	System Call Kill .....	4-52
4.11.2	Shortest-Job-First (SJF) Scheduling.....	4-24	4.21	Process Management in Multi-core OS.....	4-52
4.11.2(A)	SJF Algorithm.....	4-25	4.22	University Questions and Answers .....	4-53
4.11.3	Round Robin (RR) Scheduling .....	4-27	<b>UNIT - V</b>		
4.11.3(A)	Round Robin Algorithm .....	4-29	<hr/>		
4.11.4	SJF with Preemption Scheduling.....	4-32	<b>Chapter 5 : Synchronization and Concurrency Control</b>		
4.11.5	Priority Scheduling.....	4-33	<b>5-1 to 5-25</b>		
4.12	Interprocess Communication (Principle of Concurrency) .....	4-39	<hr/>		
4.12.1	Basic Concepts of Interprocess Communication and Synchronization.....	4-40	5.1	Interprocess Communication (Principle of Concurrency).....	5-1
4.12.2	Race Condition.....	4-40	5.1.1	Basic Concepts of Interprocess Communication and Synchronization.....	5-2
4.12.3	Critical Section.....	4-41	5.1.2	Race Condition.....	5-2
4.12.4	Mutual Exclusion.....	4-41	5.1.3	Critical Section.....	5-3
4.13	Mutual Exclusion (Software Approach) .....	4-41	5.1.4	Mutual Exclusion .....	5-3
4.14	Mutual Exclusion (Hardware Approach).....	4-43	5.2	Mutual Exclusion (Software Approach) .....	5-3
4.14.1	Disabling Interrupts.....	4-43	5.3	Mutual Exclusion (Hardware Approach).....	5-5
4.14.2	Mutual Exclusion using Machine Instructions....	4-43	5.3.1	Disabling Interrupts.....	5-5
4.15	Semaphore .....	4-43	5.3.2	Mutual Exclusion using Machine Instructions.....	5-5
4.15.1	Mutual Exclusion using Semaphore .....	4-44	5.4	Semaphore.....	5-6
4.16	IPC Problems (Classical Problems of Synchronization) .....	4-45	5.4.1	Mutual Exclusion using Semaphore .....	5-6
4.16.1	Producer/Consumer Problem.....	4-45	5.5	IPC Problems (Classical Problems of Synchronization).....	5-7
4.16.2	Readers / Writers Problem .....	4-46	5.5.1	Producer/Consumer Problem .....	5-7
4.16.3	The Dining Philosophers Problem .....	4-47	5.5.2	Readers / Writers Problem.....	5-8
4.17	Message Passing.....	4-47			



<p>5.5.3 The Dining Philosophers Problem ..... 5-9</p> <p>5.6 Message Passing..... 5-9</p> <p>5.7 Monitors..... 5-10</p> <p>5.7.1 Bounded Buffer Producer-Consumer Problem with Monitor..... 5-11</p> <p>5.8 Introduction to Deadlock..... 5-12</p> <p>5.8.1 The Conditions for Deadlock ..... 5-12</p> <p>5.9 Dealing with Deadlocks ..... 5-13</p> <p>5.9.1 Just Ignore the Problem Altogether..... 5-13</p> <p>5.9.2 Deadlock Detection and Recovery ..... 5-13</p> <p>5.9.3 Deadlock Avoidance..... 5-14</p> <p>5.9.3(A) Initial Denial..... 5-14</p> <p>5.9.3(B) Resources Allocation Denial (Banker’s Algorithm)..... 5-15</p> <p>5.9.3(C) C-program for Banker’s Algorithm ..... 5-20</p> <p>5.10 Deadlock Prevention ..... 5-22</p> <p>5.10.1 Attacking the Mutual Exclusion Condition ..... 5-22</p> <p>5.10.2 Attacking the Hold and Wait Condition..... 5-22</p> <p>5.10.3 Attacking the No Preemption Condition ..... 5-22</p> <p>5.10.4 Attacking the Circular Wait Condition ..... 5-23</p> <p>5.11 Concurrency Mechanism : Unix/Linux/Windows..... 5-23</p> <p>5.12 Concurrency Mechanism in Windows..... 5-23</p> <p>5.13 University Questions And Answers ..... 5-23</p>	<p>6.1.1 General Purpose Registers ..... 6-2</p> <p>6.1.2 Pointer and Index Registers..... 6-3</p> <p>6.1.3 Flag Register (EFLAG)..... 6-4</p> <p>6.1.4 Segment Registers ..... 6-5</p> <p>6.1.5 System Address Registers ..... 6-7</p> <p>6.1.6 Special 80386 Registers..... 6-7</p> <p>6.2 Basics of Memory Management .....6-10</p> <p>6.3 Monoprogramming without Swapping .....6-10</p> <p>6.4 Multiprogramming with Fixed Partitions (without Swapping) .....6-11</p> <p>6.5 Multiprogramming with Dynamic Partitions .....6-14</p> <p>6.5.1 Comparison between Internal and External Fragmentation .....6-15</p> <p>6.5.2 Comparison between Contiguous and Non -Contiguous Memory Allocation .....6-15</p> <p>6.5.3 Comparison between Fixed and Variable Sized Partitioning.....6-16</p> <p>6.6 Swapping and Virtual Memory.....6-16</p> <p>6.7 Paging (Non-Contiguous Memory) .....6-17</p> <p>6.7.1 Paging and Address Translation.....6-18</p> <p>6.7.2 Multilevel Page Tables.....6-19</p> <p>6.7.3 Demand Paging .....6-20</p> <p>6.7.4 Hardware Support for Paging .....6-21</p> <p>6.7.5 Sharing and Protection in a Paging System.....6-22</p> <p>6.7.6 Replacement Polices.....6-22</p> <p>6.7.6(A) FIFO Page Replacement .....6-23</p> <p>6.7.6(B) LRU Page Replacement .....6-26</p> <p>6.7.6(C) OPT Page Replacement .....6-29</p> <p>6.7.6(D) The Clock (CLOCK) Page Replacement Algorithm .....6-35</p>
---	--

## UNIT - VI

---

### Chapter 6 : Memory Management 6-1 to 6-44

---

<p>6.1 80386 Registers ..... 6-1</p>
--------------------------------------



6.7.6(E) The NRU Page Replacement Algorithm (LRU Approximation Algorithm).....	6-36	6.10 Segmentation.....	6-38
6.7.6(F) The Second Chance Page Replacement Algorithm.....	6-36	6.10.1 Segmentation with Paging.....	6-39
6.7.7 Belady's Anomaly .....	6-36	6.11 Thrashing .....	6-40
6.8 Design Issue for Paging .....	6-37	6.12 Memory Management in Multi-Core OS.....	6-41
6.8.1 The Working Set.....	6-37	6.13 Memory Management in Linux.....	6-41
6.8.2 Local versus Global Allocation Policies.....	6-37	6.14 Windows 2000 Memory Management .....	6-41
6.8.3 Page Size .....	6-37	6.15 Memory Management in Android.....	6-41
6.9 Handling of Page Fault.....	6-38	6.16 University Questions and Answers .....	6-42

