

Systems Programming and Operating System

(Code - 310251)

Semester VI - Computer Engineering
(Savitribai Phule Pune University)

Strictly as per the New Credit System Syllabus (2015 Course)
Savitribai Phule Pune University w.e.f. academic year 2017-2018

Dilip Kumar Sultania

B.Tech.(hons.) Computer Science and Engineering,
I.I.T. , Kharagpur.

 **Tech Knowledge**TM
Publications

PE36A Price ₹ 275/-



Systems Programming and Operating System (Code : 310251)

Dilip Kumar Sultania

Semester VI – Computer Engineering (Savitribai Phule Pune University)

Copyright © by Author. All rights reserved. No part of this publication may be reproduced, copied, or stored in a retrieval system, distributed or transmitted in any form or by any means, including photocopy, recording, or other electronic or mechanical methods, without the prior written permission of the publisher.

This book is sold subject to the condition that it shall not, by the way of trade or otherwise, be lent, resold, hired out, or otherwise circulated without the publisher's prior written consent in any form of binding or cover other than which it is published and without a similar condition including this condition being imposed on the subsequent purchaser and without limiting the rights under copyright reserved above.

First Edition : January 2011 **(For Pune University)**
First Edition : January 2018
Second Revised Edition : January 2019
Third Revised Edition : January 2020 **(TechKnowledge Publications)**

This edition is for sale in India, Bangladesh, Bhutan, Maldives, Nepal, Pakistan, Sri Lanka and designated countries in South-East Asia. Sale and purchase of this book outside of these countries is unauthorized by the publisher.

Printed at : 37/2, Ashtavinayak Industrial Estate,
Near Pari Company,
Narhe, Pune, Maharashtra State, India.
Pune – 411041

ISBN : 978-93-89684-96-4

Published by

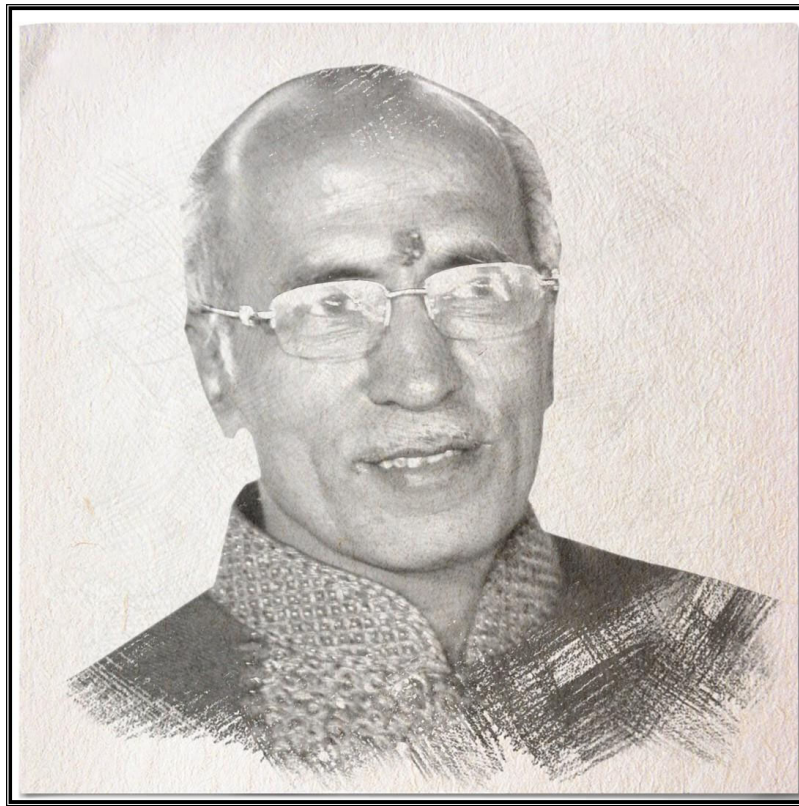
TechKnowledge Publications

Head Office : B/5, First floor, Maniratna Complex,
Taware Colony, Aranyeshwar Corner,
Pune - 411 009. Maharashtra State, India
Ph : 91-20-24221234, 91-20-24225678.
Email : info@techknowledgebooks.com,
Website : www.techknowledgebooks.com

[310251] (FID : PE36) (Book Code : PE36A)

(Book Code : PE36A)

*We dedicate this Publication soulfully and wholeheartedly,
in loving memory of our beloved founder director,
Late Shri. Pradeepji Lalchandji Lunawat,
who will always be an inspiration, a positive force and strong support
behind us.*



“My work is my prayer to God”

- Lt. Shri. Pradeepji L. Lunawat

*Soulful Tribute and Gratitude for all Your
Sacrifices, Hardwork and 40 years of Strong Vision...*

Preface

My Dear Students,

I am extremely happy to come out with this book on “**Systems Programming and Operating System**” for you. The topics within the chapters have been arranged in a proper sequence to ensure smooth flow of the subject.

A large number of solved examples have also been included. Therefore, we are sure that this book will cater all your needs for this subject.

I present this book in the loving memory of **Late Shri. Pradeepji Lunawat**, our source of inspiration and a strong foundation of “**TechKnowledge Publications**”. He will always be remembered in our heart and motivate us to achieve our milestone.

I am thankful to the staff members of TechKnowledge Publications and others for their efforts to make this book as good as it is. I have jointly made every possible efforts to eliminate all the errors in this book. However if you find any, please let me know, because that will help me to improve further.

I am also thankful to my family members and friends for patience and encouragement.

- Dilip Kumar Sultania

□□□

Syllabus

Savitribai Phule Pune University
Third Year of Computer Engineering (2015 Course)
310251 : Systems Programming and Operating System

Teaching Scheme	Credits	Examination Scheme
TH : 4 Hours/Week	04	In-Sem(Paper) : 30 Marks End-Sem(Paper) : 70 Marks

Prerequisite Courses: Fundamentals of Programming Languages(110011,110003), Data Structures (210243,210252)

Companion Course : Systems Programming and Operating System Lab (310257)

Course Objectives

- To understand basics of System Programming.
- To learn and understand data structures used in design of system software.
- To learn and understand basics of compilers and tools.
- To understand functions of operating system.
- To learn and understand process, resource and memory management.

Course Outcomes

On completion of the course, student will be able to–

- Analyze and synthesize system software
- Use tools like LEX & YACC.
- Implement operating system functions.

Course Contents

UNIT I : Introduction

(09 Hours)

Introduction: Components of System Software: Text editors, Loaders, Assemblers, Macro processors, Compilers, Debuggers. Machine Structure, Machine language and Assembly Language. Assemblers: General design procedure, design of two pass assembler (Refer Chapter 1)

UNIT II : Macro Processor, Linker and Loader

(09 Hours)

Macro Processor : Macro instructions, Features of macro facility, Design of two-pass, single pass and nested macro processor. Loaders: Loader schemes: Compile and go, General Loader Scheme, Absolute loaders, subroutine linkages, relocating loaders, direct linking loaders, overlay structure. Design of an absolute loader, Design of direct linking loader. Linkers: Relocation and linking concepts, Design of linker, self relocating programs, Static and dynamic link libraries, use of call back functions. Case Study: Loading phases using Java.

(Refer Chapters 2, 3 and 4)

UNIT III : Language Translator**(09 Hours)**

Role of lexical analysis - parsing & Token, patterns and Lexemes & Lexical Errors, regular definitions for the language constructs & strings, sequences, Comments & Transition diagram for recognition of tokens, reserved words and identifiers, examples Introduction to Compilers and Interpreters: General Model of Compiler, Program interpretation, Comparison of compiler and Interpreter, Use of Interpreter and components of Interpreter. Case Study: Overview of LEX and YACC specification and features. **(Refer Chapter 5)**

UNIT IV : Operating Systems**(09 Hours)**

Operating Systems: Introduction to different types of operating Real Time Operating Systems, System Components, OS services, System structure- Layered Approach. Process Management: Process Concept- Process states, Process control block, Threads, Process Scheduling: Types of process schedulers, Types of scheduling: Preemptive, Non preemptive. Scheduling algorithms: FCFS, SJF, RR, Priority, Deadlocks: Methods of handling deadlocks, Deadlock prevention, avoidance and detection, Recovery from deadlocks. Case Study: Process Management in multi-cores OS. **(Refer Chapter 6)**

UNIT V : Memory Management**(09 Hours)**

Memory management: Review of Programming Model of Intel 80386, Contiguous and non-contiguous, Swapping, Paging, Segmentation, Segmentation with Paging. Virtual Memory: Background, Demand paging, Page replacement scheme- FIFO, LRU, Optimal, Thrashing. Case Study: Memory Management in multi-cores OS. **(Refer Chapter 7)**

UNIT VI : I/O Management**(09 Hours)**

I/O Management: I/O Devices, Organization of I/O function, I/O Buffering, Disk Scheduling- Disk Scheduling policies like FIFO, LIFO, STTF, SCAN, C-SCAN. File Management: Concept, Access methods, Directory Structure, Protection, File System implementation, Directory Implementation, Allocation methods, Free Space management. Case Study: I/O and File Management in multi-cores OS. Case Study: Light weight and heavy weight OS: Linux, Tizen. **(Refer Chapter 8)**

□□□

Savitribai Phule Pune University
Third Year of Computer Engineering (2015 Course)
310257 : Systems Programming and Operating System Lab

Teaching Scheme	Credits	Examination Scheme
PR : 4 Hours/Week	02	TW : 25 Marks
		PR : 50 Marks

Companion Course : Systems Programming and Operating System (310251)

Course Objectives

- To implement basic language translator by using various needed data structures
- To implement basic Macroprocessor
- To design and implement Dynamic Link Libraries
- To implement scheduling schemes

Course Outcomes

On completion of the course, student will be able to–

- Understand the internals of language translators
- Handle tools like LEX & YACC.
- Understand the Operating System internals and functionalities with implementation point of view

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface etc), University syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and **handwritten write-up** of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, Design, test cases, conclusion/analysis. **Program codes with sample output of all performed assignments are to be submitted as softcopy.**

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Use of DVD containing students programs maintained by lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

Guidelines for Assessment

Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of student. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation.

So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. Encourage students for appropriate use of Hungarian notation, proper indentation and comments. Use of open source software is to be encouraged. In addition to these, instructor may assign one real life application in the form of a mini-project based on the concepts learned.

Set of suggested assignment list is provided in groups- A, B, C, D (All Compulsory)

Operating System recommended : 64-bit Open source Linux or its derivative

Programming tools recommended : Eclipse IDE

Suggested List of Laboratory Assignments	
Group A	
1.	Design suitable data structures and implement pass-I of a two-pass assembler for pseudo-machine in Java using object oriented feature. Implementation should consist of a few instructions from each category and few assembler directives.
2.	Implement Pass-II of two pass assembler for pseudo-machine in Java using object oriented features. The output of assignment-1 (intermediate file and symbol table) should be input for this assignment.
3.	Design suitable data structures and implement pass-I of a two-pass macro-processor using OOP features in Java.
4.	Write a Java program for pass-II of a two-pass macro-processor. The output of assignment-3 (MNT, MDT and file without any macro definitions) should be input for this assignment.
Group B	
1.	Write a program to create Dynamic Link Library for any mathematical operation and write an application program to test it. (Java Native Interface / Use VB or VC++).
2.	Write a program using Lex specifications to implement lexical analysis phase of compiler to generate tokens of subset of 'Java' program.
3.	Write a program using Lex specifications to implement lexical analysis phase of compiler to count no. of words, lines and characters of given input file.
4.	Write a program using YACC specifications to implement syntax analysis phase of compiler to validate type and syntax of variable declaration in Java.
5.	Write a program using YACC specifications to implement syntax analysis phase of compiler to recognize simple and compound sentences given in input file.
Group C	
1.	Write a Java program (using OOP features) to implement following scheduling algorithms: FCFS , SJF (Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive)
2.	Write a Java program to implement Banker's Algorithm
3.	Implement UNIX system calls like ps, fork, join, exec family, and wait for process management (use shell script/ Java/ C programming).
4.	Study assignment on process scheduling algorithms in Android and Tizen.
Group D	
1.	Write a Java Program (using OOP features) to implement paging simulation using 1. Least Recently Used (LRU) 2. Optimal algorithm

□□□

UNIT I

Chapter 1 : Introduction to System Programming

1-1 to 1-34

Syllabus : Introduction : Components of System Software: Text editors, Loaders, Assemblers, Macro processors, Compilers, Debuggers. Machine Structure, Machine language and Assembly Language. Assemblers: General design procedure, design of two pass assembler.

✓	Syllabus Topic : Components of System Software	1-1
1.1	Software Tools for Program Development	1-1
✓	Syllabus Topic : Text Editors.....	1-1
1.1.1	Editor (Oct. 2016).....	1-1
1.1.1(a)	Editor Structure.....	1-2
✓	Syllabus Topic : Debuggers	1-2
1.1.2	Debug Monitor	1-2
1.1.3	Programming Environment.....	1-2
1.1.4	User Interfaces.....	1-2
✓	Syllabus Topic : Loaders.....	1-3
1.2	Basic Functions of Loaders	1-3
✓	Syllabus Topic : Assemblers	1-4
1.3	Assembly Language Programming	1-4
✓	Syllabus Topic : Macro Processors.....	1-4
1.4	Macro Processor	1-4
✓	Syllabus Topic : Compilers.....	1-5
1.5	Compiler	1-5
1.5.1	An Interpreter (Dec. 2013)	1-5
1.5.2	Difference between Interpreter and Compiler (Aug. 2015, Oct. 2016, May 2017)	1-6
1.5.3	A Cross Compiler	1-6
1.5.4	A Bootstrap Compiler	1-6
✓	Syllabus Topic : Machine Structure.....	1-6
1.6	The Von Neumann Architecture	1-6
1.6.1	Input Unit	1-7
1.6.2	Output Unit	1-7
1.6.3	Arithmetic and Logic Unit (ALU)	1-7
1.6.4	Control Unit.....	1-7
1.6.5	Memory Unit	1-7
1.6.6	Key Features of a Von Neumann Machine	1-7
1.6.7	Basic Structure of the CPU	1-7
1.6.8	Execution of a Program by Von Neumann Machine	1-8
✓	Syllabus Topic : Machine Language and Assembly Language	1-10
1.7	Assembly Language Programming	1-10
1.7.1	Elements of Assembly Language	1-10
1.7.2	Assembly Program to Machine Language Program ..	1-11
1.7.3	Assembly Language Statements (Dec. 2016, May 2017)	1-12

1.7.4	Literals and Constants	1-13
✓	Syllabus Topic : Assemblers - General Design Procedure	1-14
1.8	Assembler and Related Programs (Dec. 2013).....	1-14
1.8.1	Load and Go Assembler	1-14
1.8.2	One-pass Assembler	1-14
1.8.2(a)	One-pass Assembler with Forward References	1-15
1.8.2(b)	Design of Single Pass Assembler.....	1-15
1.8.3	Two Pass Assembler	1-16
1.9	Advanced Assembler Directives (May 2014, Aug. 2015, Oct. 2016)	1-16
✓	Syllabus Topic : Design of Two Pass Assembler	1-17
1.10	Design of Two Pass Assembler (May 2013, Dec. 2015)	1-17
1.10.1	Intermediate Code	1-20
1.10.1(a)	Variants of Intermediate Code	1-21
1.10.2	Algorithm for Pass I (May 2013)	1-22
1.10.3	Flowchart for Pass I	1-25
1.10.4	C-Program for Pass I	1-25
1.10.5	Pass II of the Assembler (May 2014)	1-29
1.10.6	C-Program for Pass II	1-29
1.10.7	Error Reporting	1-33
1.11	Exam Pack (University Questions)	1-34

UNIT II

Chapter 2 : Macro Processor

2-1 to 2-19

Syllabus : Macro Processor : Macro instructions, Features of macro facility, Design of two-pass, single pass and nested macro processor.

2.1	Macro Definition (May 2014)	2-1
2.2	Defining a Macro.....	2-2
2.2.1	Calling a Macro (May 2014)	2-2
2.2.2	Macro Expansion (May 2013, May 2014).....	2-2
2.2.3	Macro with Keyword Parameters (Dec. 2015).....	2-3
2.2.4	Macro with Mixed Parameters	2-5
2.2.5	Other Uses of Parameters	2-5
✓	Syllabus Topic : Nested Macro Processor.....	2-5
2.3	Nested Macro Calls (Dec. 2014, May 2015, May 2016)	2-5
2.3.1	Nested Macro Definition	2-6
✓	Syllabus Topic : Macro Instructions Features of Macro Facility.....	2-6
2.4	Advanced Macro Facilities (May 2013, Dec. 2015, May 2017)	2-6
2.4.1	Expansion Time Variables (EV) (Dec. 2014, May 2015, Dec. 2015)	2-7
✓	Syllabus Topic : Design of Two-pass Macro-processor	2-8

2.5	Design of Macro Processor (May 2013, May 2014)	2-8
2.5.1	Issues Related to the Design of a Simple Macro Preprocessor	2-8
2.5.2	Databases used in Pass-1 of 2 Pass Macro Processor (Dec. 2013)	2-9
2.5.3	Flow Chart for Pass 1	2-9
2.5.4	Databases used in Pass-2 of 2 Pass Macro Processor (Dec. 2013)	2-9
2.5.5	Flow Chart for Pass-2	2-10
✓	Syllabus Topic : Design of Single Pass Macro Processor	2-10
2.5.6	A Simple One-Pass Macro Processor	2-10
2.5.7	C-Program for One-pass Macro Processor	2-11
2.6	Handling of Nested Macro Calls (Dec. 2013, May 2014)	2-15
2.7	Handling of Nested Macro Declaration	2-17
2.8	Exam Pack (University Questions)	2-19

Chapter 3 : Loaders 3-1 to 3-16

Syllabus : Loaders : Loader schemes: Compile and go, General Loader Scheme, Absolute loaders, subroutine linkages, relocating loaders, direct linking loaders, overlay structure. Design of an absolute loader, Design of direct linking loader.

3.1	Basic Functions of a Loader (May 2014, Dec. 2014, Aug. 2015, Dec. 2015)	3-1
✓	Syllabus Topic : Loader Schemes	3-2
3.2	Loading Schemes (May 2014, Dec. 2014, Dec. 2015, Dec. 2016)	3-2
✓	Syllabus Topic : Compile and go	3-2
3.2.1	Compile-and-go Loaders (Dec. 2014, Dec. 2015, Oct. 2016)	3-2
✓	Syllabus Topic : General Loader Scheme	3-3
3.2.2	General Loader Scheme	3-3
✓	Syllabus Topic : Absolute Loaders	3-3
3.2.3	Absolute Loader (Aug. 2015)	3-3
✓	Syllabus Topic : Subroutine Linkages	3-3
3.2.4	Subroutine Linkage (Dec. 2013)	3-3
✓	Syllabus Topic : Relocating Loaders	3-4
3.2.5	Relocating Loader (Dec. 2013)	3-4
✓	Syllabus Topic : Direct Linking Loaders	3-5
3.2.6	Direct Linking Loader (May 2013)	3-5
✓	Syllabus Topic : Design of an Absolute Loader	3-7
3.3	Design of an Absolute Loader (Dec. 2013)	3-7
✓	Syllabus Topic : Design of Direct Linking Loader	3-7
3.4	Design of Direct Linking Loader (May 2015)	3-7
3.4.1	Format of Databases of a Direct Linking Loader	3-10
3.4.2	Flowchart for Pass I	3-11
3.4.3	Flowchart for Pass II	3-11
3.5	Implementation of MS DOS Linker (May 2014)	3-12
3.5.1	Data Structures and Algorithm	3-12

✓	Syllabus Topic : Overlay Structure	3-13
3.6	Overlay Structure	3-13
3.7	Software Tools for Program Development	3-14
3.7.1	Editor (Oct. 2016)	3-14
3.7.1(a)	Editor Structure	3-15
3.7.2	Debug Monitor	3-15
3.7.3	Programming Environment	3-15
3.7.4	User Interfaces	3-15
3.8	Exam Pack (University Questions)	3-16

Chapter 4 : Linkers 4-1 to 4-6

Syllabus : Relocation and linking concepts, Design of linker, Self relocating programs, Static and dynamic link libraries, use of call back functions. Case Study : Loading phases using Java.

✓	Syllabus Topic : Relocation and Linking Concepts	4-1
4.1	Relocation and Linking Concept	4-1
4.1.1	Linking (May 2013, May 2014, Dec. 2014)	4-1
4.1.2	Relocation (May 2013)	4-1
4.1.3	Object Module	4-2
✓	Syllabus Topic : Self Relocating Programs	4-2
4.2	Self Relocating Programs (Aug. 2015)	4-2
✓	Syllabus Topic : Static and Dynamic Link Libraries	4-2
4.3	Static and Dynamic Link Libraries	4-2
4.3.1	Static Linking	4-2
4.3.2	Dynamic Linking (Oct. 2016)	4-2
4.3.3	Dynamic Link Library (DLL)	4-3
✓	Syllabus Topic : Use of Call Back Functions	4-3
4.3.4	Call Back Functions	4-3
4.3.5	Difference between Callback Function and Normal Function	4-4
4.4	Dynamic Linking with and without Import	4-4
4.4.1	Using the DLL (with an import library)	4-4
4.4.2	Using DLL (without an Important Library)	4-5
✓	Syllabus Topic : Case Study : Loading Phases using Java	4-5
4.5	Loading Phases using Java	4-5
4.6	Exam Pack (University Questions)	4-6

UNIT III

Chapter 5 : Language Translator 5-1 to 5-45

Syllabus : Role of lexical analysis -parsing & Token, patterns and Lexemes & Lexical Errors, regular definitions for the language constructs & strings, sequences, Comments & Transition diagram for recognition of tokens, reserved words and identifiers, examples. Introduction to Compilers and Interpreters: General Model of Compiler, Program interpretation, Comparison of compiler and Interpreter, Use of Interpreter and components of Interpreter. Case Study: Overview of LEX and YACC specification and features.

✓	Syllabus Topic : Introduction to Compilers and Interpreters	5-1
5.1	Compiler	5-1

5.1.1	An Interpreter (Dec. 2013)	5-2
✓	Syllabus Topic : Comparison of Compilers and Interpreters	5-2
5.1.2	Difference between Interpreter and Compiler (Aug. 2015, Oct. 2016, May 2017)	5-2
5.1.3	A Cross Compiler	5-2
5.1.4	A Bootstrap Compiler	5-3
5.2	Deterministic Finite Automata (DFA).....	5-3
5.2.1	Definition of a DFA (May 2013).....	5-3
5.2.1(a)	Representation of a DFA.....	5-3
5.2.1(b)	Designing a DFA	5-4
✓	Syllabus Topic : Regular Definitions for the Language Constructs & Strings, Sequences, Comments & Transition Diagram for Recognition of Tokens, Reserved Words and Identifiers, Examples	5-6
5.2.2	Regular Expression (May 2013).....	5-6
5.2.3	RE to DFA	5-7
5.2.3(a)	NFA to DFA	5-8
5.2.3(b)	Finding ϵ -Closures.....	5-8
5.2.4	From a Regular Expression to DFA (Direct Approach)	5-10
✓	Syllabus Topic : Token, Pattern and Lexemes and Lexical Errors.....	5-19
5.3	Token, Pattern and Lexemes and Lexical Errors	5-19
✓	Syllabus Topic : General Model of Compiler.....	5-19
5.4	General Model of Compiler (Aug. 2015, May 2017)	5-19
✓	Syllabus Topic : Role of Lexical Analysis.....	5-20
5.4.1	Lexical Analysis (Dec. 2016, May 2017).....	5-20
5.4.2	Syntax Analysis (Dec. 2016, May 2017).....	5-25
5.4.3	Semantic Analysis (May 2017).....	5-26
✓	Syllabus Topic : Compilation of Expressions	5-26
5.4.4	Intermediate Code Generation	5-26
5.4.5	Code Optimization	5-27
5.4.6	Code Generation	5-27
5.5	Intermediate Code for Array	5-31
5.6	Compilation of Control Structure	5-32
5.6.1	if Statement	5-32
5.6.2	while Statement	5-32
5.6.3	function and procedure call	5-33
5.7	Code Optimization (May 2013, Oct. 2016, Dec. 2016)	5-33
5.7.1	Machine Dependent Optimization	5-35
5.8	Issues in the Design of a Code Generator	5-37
5.8.1	Input to Code Generator.....	5-37
5.8.2	Target Program	5-38
5.8.3	Memory Management	5-38
5.8.4	Instruction Selection	5-38
5.8.5	Register Allocation.....	5-38
5.8.6	Choice of Evaluation Order	5-38
5.8.7	Approaches to Code Generation :	5-38

5.9	Code Generation Algorithm (for + operation)	5-38
✓	Syllabus Topic : Use of Interpreter and Components of Interpreter	5-39
5.10	Use of Interpreter and Components of Interpreter....	5-39
✓	Syllabus Topic : Overview of LEX and YACC Specification and Features	5-39
5.11	Overview of LEX and YACC Specification and Features (May 2013, May 2014, May 2015, May 2016)	5-39
5.11.1	Lex (May 2013).....	5-40
5.11.1(a)	Lex Examples	5-41
5.11.2	Yacc (May 2013).....	5-42
5.11.3	Running Lex and Yacc.....	5-43
5.12	Exam Pack (University Questions)	5-44

UNIT IV

Chapter 6 : Operating Systems

6-1 to 6-47

Syllabus : Operating Systems : Introduction to different types of operating system, Real Time Operating Systems, System Components, OS services, System structure- Layered Approach.

Process Management: Process Concept- Process states, Process control block, Threads, Process Scheduling: Types of process schedulers, Types of scheduling: Preemptive, Non preemptive. Scheduling algorithms: FCFS, SJF, RR, Priority,

Deadlocks : Methods of handling deadlocks, Deadlock prevention, avoidance and detection, Recovery from deadlocks.

Case Study : Process Management in multi-cores OS.

✓	Syllabus Topic : OS Services	6-1
6.1	Evolution of O.S. Functions (May 2014, Dec. 2014)	6-1
6.1.1	Functions of an Operating System (May 2013, Dec. 2013, Dec. 2016)	6-2
6.1.1(a)	The Operating System as a User/Computer Interface	6-2
6.1.1(b)	The Operating System as Resource Manager	6-3
6.2	Evolution of Operating Systems (Dec. 2016)	6-3
6.2.1	Serial Processing	6-3
6.2.2	Batch Processing	6-3
6.2.3	Multiprogramming	6-4
✓	Syllabus Topic : Different Types of Operating System	6-5
6.3	Types of Operating System (May 2014).....	6-5
6.3.1	Batch Operating System	6-5
6.3.2	Multiprogramming Operating System	6-6
✓	Syllabus Topic : Real Time Operating System	6-6
6.3.3	Real-time System (Dec. 2015)	6-6
6.3.4	Network Operating System	6-6
6.3.5	Distributed Operating System	6-6
✓	Syllabus Topic : System Components.....	6-7
6.4	Operating System Components.....	6-7
6.4.1	Processes	6-7
6.4.2	Files	6-7

6.4.3	System Calls (Dec. 2013, Dec. 2014, May 2015, Dec. 2015, May 2016).....	6-7	6.13	Mutual Exclusion (Hardware Approach)	6-31
6.4.4	Command Interpreter	6-8	6.13.1	Disabling Interrupts	6-31
6.4.5	Signals.....	6-8	6.13.2	Mutual Exclusion using Machine Instructions	6-31
✓	Syllabus Topic : System Structure	6-9	6.14	Semaphore	6-32
6.5	Operating System Structure	6-9	6.14.1	Mutual Exclusion using Semaphore	6-32
6.5.1	Monolithic System	6-9	6.15	IPC Problems (Classical Problems of Synchronization)	6-32
✓	Syllabus Topic : Layered Approach	6-10	6.15.1	Producer/Consumer Problem (Dec. 2014, May 2015, May 2017)	6-32
6.5.2	Layered Systems.....	6-10	6.15.2	Readers / Writers Problem (May 2016)	6-33
6.5.3	Virtual Machine.....	6-10	6.15.3	The Dining Philosophers Problem (Dec. 2014, May 2015, May 2016)	6-34
6.5.4	Client-Server Model.....	6-10	6.16	Message Passing.....	6-34
✓	Syllabus Topic : Process Management - Process Concept	6-11	6.17	Monitors	6-35
6.6	Process (Dec. 2015).....	6-11	6.17.1	Bounded Buffer Producer-Consumer Problem with Monitor	6-35
6.7	Process Control.....	6-11	6.18	Introduction to Deadlock (May 2013, Dec. 2013, May 2014, May 2016)	6-36
✓	Syllabus Topic : Process States.....	6-11	6.18.1	The Conditions for Deadlock (Dec. 2015)	6-36
6.7.1	Process States (May 2013, Dec. 2014, May 2015, May 2016, Dec. 2016, May 2017).....	6-11	✓	Syllabus Topic : Methods of Handling Deadlocks	6-37
✓	Syllabus Topic : Process Control Block	6-12	6.19	Dealing with Deadlocks	6-37
6.7.2	Process Implementation (May 2013, Dec. 2016).....	6-12	6.19.1	Just Ignore the Problem Altogether (Dec. 2013)	6-37
✓	Syllabus Topic : Threads.....	6-12	✓	Syllabus Topic : Deadlock Detection, Recovery from Deadlock	6-37
6.8	Threads (Dec. 2014, May 2015, Dec. 2015).....	6-12	6.19.2	Deadlock Detection and Recovery	6-37
6.8.1	Benefits of Threads	6-13	✓	Syllabus Topic : Deadlock Avoidance	6-38
✓	Syllabus Topic : Process Scheduling.....	6-13	6.19.3	Deadlock Avoidance (May 2013, May 2014)	6-38
6.9	Process Scheduling (May 2016)	6-13	6.19.3(a)	Initial Denial	6-38
✓	Syllabus Topic : Types of Process Schedulers	6-14	6.19.3(b)	Resources Allocation Denial (Banker's Algorithm) (Dec. 2015, May 2017)	6-38
6.9.1	Types of Schedulers.....	6-14	6.19.3(c)	C-program for Banker's Algorithm	6-42
✓	Syllabus Topic : Types of Scheduling : Preemptive, Non Preemptive.....	6-14	✓	Syllabus Topic : Deadlock Prevention	6-43
6.10	Scheduling Methods (May 2014, Dec. 2015).....	6-14	6.20	Deadlock Prevention (Dec. 2013, May 2014, May 2016)	6-43
✓	Syllabus Topic : Scheduling Algorithm - FCFS	6-15	6.20.1	Attacking the Mutual Exclusion Condition.....	6-43
6.10.1	First-Come-First-Served (FCFS) Scheduling (May 2017)	6-15	6.20.2	Attacking the Hold and Wait Condition	6-43
6.10.1(a)	FCFS Algorithm	6-16	6.20.3	Attacking the No Preemption Condition.....	6-43
✓	Syllabus Topic : Scheduling Algorithm - SJF	6-17	6.20.4	Attacking the Circular Wait Condition	6-44
6.10.2	Shortest-Job-First (SJF) Scheduling (May 2017).....	6-17	6.21	Pipes	6-44
6.10.2(a)	SJF Algorithm	6-18	6.22	System Calls	6-45
✓	Syllabus Topic : Scheduling Algorithm - RR	6-20	6.22.1	System Call Signal.....	6-45
6.10.3	Round Robin (RR) Scheduling (May 2013)	6-20	6.22.2	System Call Kill.....	6-45
6.10.3(a)	Round Robin Algorithm	6-21	✓	Syllabus Topic : Case Study : Process Management in Multi-core OS	6-45
6.10.4	SJF with Preemption Scheduling	6-23	6.23	Process Management in Multi-core OS	6-45
✓	Syllabus Topic : Scheduling Algorithm - Priority	6-24	6.24	Exam Pack (University Questions)	6-46
6.10.5	Priority Scheduling.....	6-24			
6.11	Interprocess Communication (Principle of Concurrency)(May 2014, Dec. 2015, May 2016).....	6-28			
6.11.1	Basic Concepts of Interprocess Communication and Synchronization.....	6-29			
6.11.2	Race Condition	6-29			
6.11.3	Critical Section (May 2013)	6-30			
6.11.4	Mutual Exclusion	6-30			
6.12	Mutual Exclusion (Software Approach)	6-30			

UNIT V

Chapter 7 : Memory Management 7-1 to 7-30

Syllabus : Memory management : Review of Programming Model of Intel 80386, Contiguous and non-contiguous, Swapping, Paging, Segmentation, Segmentation with Paging. Virtual Memory: Background, Demand paging, Page replacement scheme- FIFO, LRU, Optimal, Thrashing.

Case Study : Memory Management in multi-cores OS.

✓	Syllabus Topic : Review of Programming Model of Intel 80386	7-1
7.1	80386 Registers	7-1
7.1.1	General Purpose Registers	7-2
7.1.2	Pointer and Index Registers	7-3
7.1.3	Flag Register (EFLAG)	7-4
7.1.4	Segment Registers	7-5
7.1.5	System Address Registers	7-5
7.1.6	Special 80386 Registers	7-5
7.2	Basics of Memory Management (Dec. 2013)	7-7
7.3	Monoprogramming without Swapping	7-8
✓	Syllabus Topic : Contiguous Memory Allocation	7-8
7.4	Multiprogramming with Fixed Partitions (without Swapping) (Dec. 2014, May 2015, Dec. 2015, May 2016, Dec. 2016, May 2017)	7-8
7.5	Multiprogramming with Dynamic Partitions (May 2016)	7-10
✓	Syllabus Topic : Swapping, Virtual Memory : Background	7-11
7.6	Swapping and Virtual Memory (May 2013, Dec. 2013, May 2014)	7-11
✓	Syllabus Topic : Paging, Non-contiguous Memory Allocation	7-12
7.7	Paging (Non-contiguous Memory)	7-12
7.7.1	Paging and Address Translation	7-12
7.7.2	Multilevel Page Tables	7-14
✓	Syllabus Topic : Demand Paging	7-14
7.7.3	Demand Paging (May 2014, Dec. 2014, May 2015, Dec. 2015, Dec. 2016)	7-14
7.7.4	Hardware Support for Paging (Dec. 2015)	7-15
7.7.5	Sharing and Protection in a Paging System	7-16
✓	Syllabus Topic : Page Replacement Scheme	7-16
7.7.6	Replacement Policies (May 2013, Dec. 2013, Dec. 2015, May 2016, Dec. 2016, May 2017)	7-16
✓	Syllabus Topic : Page Replacement Scheme – FIFO	7-17
7.7.6(a)	FIFO Page Replacement (Dec. 2013, May 2014)	7-17
✓	Syllabus Topic : Page Replacement Scheme - LRU	7-19
7.7.6(b)	LRU Page Replacement (Dec. 2013, May 2014, May 2016, Dec. 2016, May 2017)	7-19

✓	Syllabus Topic : Page Replacement Scheme - Optimal	7-21
7.7.6(c)	OPT Page Replacement (May 2014)	7-21
7.7.6(d)	The Clock (CLOCK) Page Replacement Algorithm	7-25
7.7.6(e)	The NRU Page Replacement Algorithm (LRU Approximation Algorithm)	7-26
7.7.6(f)	The Second Chance Page Replacement Algorithm (May 2014)	7-26
7.7.7	Belady's Anomaly	7-26
7.8	Design Issue for Paging (Dec. 2014, May 2015, May 2016, Dec. 2016)	7-26
7.8.1	The Working Set	7-26
7.8.2	Local versus Global Allocation Policies	7-26
7.8.3	Page Size	7-27
7.9	Handling of Page Fault	7-27
✓	Syllabus Topic : Segmentation	7-28
7.10	Segmentation (Dec. 2015, May 2016, Dec. 2016)	7-28
✓	Syllabus Topic : Segmentation with Paging	7-28
7.10.1	Segmentation with Paging	7-28
✓	Syllabus Topic : Thrashing	7-29
7.11	Thrashing	7-29
✓	Syllabus Topic : Memory Management in Multi-Core OS	7-29
7.12	Memory Management in Multi-Core OS	7-29
7.13	Exam Pack (University Questions)	7-29

UNIT VI

Chapter 8 : Input and Output, File System 8-1 to 8-35

Syllabus : I/O Management : I/O Devices, Organization of I/O function, I/O Buffering, Disk Scheduling- Disk Scheduling policies like FIFO, LIFO, STTF, SCAN, C-SCAN.

File Management: Concept, Access methods, Directory Structure, Protection, File System implementation, Directory Implementation, Allocation methods, Free Space management.

Case Study: I/O and File Management in multi-cores OS

Case Study: Light weight and heavy weight OS: Linux, Tizen.

8.1	Overview	8-1
8.2	Principles of I/O Hardware	8-1
✓	Syllabus Topic : I/O Devices	8-1
8.2.1	I/O Devices	8-1
8.2.2	Device Controller	8-2
✓	Syllabus Topic : Organization of I/O Function	8-2
8.3	Techniques for Performing I/O (Organization of I/O Functions)	8-2
8.3.1	Programmed I/O (May 2017)	8-2
8.3.1(a)	Input/output Addressing (May 2017)	8-3
8.3.2	Interrupt Driven I/O (May 2017)	8-4
8.3.2(a)	Comparison between Programmed and Interrupt Driven Input/Output	8-4



8.3.2(b)	Interrupt Processing (Dec. 2013).....	8-5	✓	Syllabus Topic : Access Methods	8-24
8.3.2(c)	Difference between Subroutine and Interrupt Service Routine.....	8-5	8.8.4	Access Methods (File Organization) (May 2016, May 2017)	8-24
8.3.2(d)	Types of Interrupts	8-6	8.8.4(a)	The Pile.....	8-24
8.3.3	DMA (Direct Memory Access) (May 2017).....	8-6	8.8.4(b)	Sequential File	8-24
8.3.3(a)	DMA Data Transfer Modes	8-7	8.8.4(c)	Indexed File.....	8-25
8.4	Principles of I/O Software (Operating System Design Issues) (May 2013, Dec. 2013, May 2014, Dec. 2014, May 2015, May 2016, Dec. 2016, May 2017)	8-8	8.8.4(d)	Hashed (Direct) File Organization.....	8-25
8.4.1	Interrupt Handlers.....	8-8	8.8.4(e)	Indexed Sequential File	8-26
8.4.2	Device Drivers	8-8	8.8.4(f)	Indexing and Hashing Comparison.....	8-26
8.4.3	Device-Independent I/O Software	8-9	✓	Syllabus Topic : Directory Structure	8-27
8.4.4	User-Space I/O Software.....	8-9	8.9	Directories (May 2014, Dec. 2014, Dec. 2015, May 2016, Dec. 2016)	8-27
✓	Syllabus Topic : I/O Buffering	8-9	8.9.1	Path Names	8-27
8.5	I/O Buffering	8-9	✓	Syllabus Topic : File System Implementation, Allocation Methods	8-27
8.5.1	Single Buffer.....	8-10	8.10	File System Implementation (Dec. 2015, May 2017).....	8-27
8.5.2	Double Buffer	8-10	8.10.1	Contiguous Allocation.....	8-28
8.5.3	Circular Buffer	8-10	8.10.2	Linked Allocation	8-28
8.6	Magnetic Disk (May 2016).....	8-10	8.10.3	Indexed Allocation	8-29
8.6.1	RAID (Redundant Array of Independent Disks) (Dec. 2014, Dec. 2015, May 2016).....	8-11	8.10.3(a)	Multi-level Indexing.....	8-29
8.6.2	Disk Cache.....	8-13	✓	Syllabus Topic : Free Space Management	8-30
✓	Syllabus Topic : Disk Scheduling	8-13	8.11	Disk Space Management (May 2013).....	8-30
8.7	Disk Scheduling.....	8-13	8.11.1	Linked List (May 2017)	8-30
8.7.1	First-come-First-Served (FCFS) Scheduling.....	8-13	8.11.2	Bit Map (May 2017)	8-30
✓	Syllabus Topic : Disk Scheduling Policy - SSTF	8-13	✓	Syllabus Topic : Directory Implementation	8-31
8.7.2	Shortest-Seek-Time-First (SSTF) Scheduling	8-14	8.12	Implementing Directories (File System) (Dec. 2015)	8-31
✓	Syllabus Topic : Disk Scheduling Policy - SCAN	8-15	8.12.1	File System in CP/M.....	8-31
8.7.3	Scan Scheduling	8-15	8.12.2	File System in MS-DOS	8-31
8.7.3(a)	Advantages and Disadvantages of SCAN Algorithm	8-16	8.12.3	File System in UNIX (Dec. 2015).....	8-32
✓	Syllabus Topic : Disk Scheduling Policy - C-SCAN	8-17	8.13	Shared Files (Acyclic Graph Directories) (Dec. 2016)	8-33
8.7.4	Circular Scan (C-scan) Scheduling.....	8-17	8.14	Record Blocking	8-33
8.7.4(a)	Advantage and Disadvantages of C-Scan Algorithm	8-17	✓	Syllabus Topic : Light Weight and Heavy Weight OS : Linux, Tizen	8-34
✓	Syllabus Topic : File Management-Concept	8-22	8.15	Light Weight and Heavy Weight OS	8-34
8.8	Files (May 2013, Dec. 2013)	8-22	8.16	Exam Pack (University Questions)	8-34
8.8.1	File Operations (May 2014, Dec. 2014, May 2015, Dec. 2015, May 2016, Dec. 2016).....	8-23	•	Lab Manual	L1- to L-37
8.8.2	File Types (May 2017)	8-23	•	Appendix-A : Solved University Question Papers of April 2018 and May 2018	A-1 to A-39
8.8.3	File Structures (May 2013, Dec. 2013, May 2014)	8-24	•	Appendix-B : Solved University Question Papers of Dec. 2018, April 2019 and May 2019	B-1 to B-10

List of Practicals

Practical No.	Name of the Program	Page No.
1.	Design suitable data structures and implement Pass-I of a two-pass assembler for 8 bit microprocessor/pseudo-machine. Implementation should consist of a few instructions from each category and few assembler directives.	L-1
2.	Implement Pass-II of a two-pass assembler for 8-bit microprocessor/pseudo-machine. The output of assignment-I (intermediate file and symbol table) should be input for this assignment.	L-9
3.	Design suitable data structures and implement pass-I of a two-pass macro-processor using OOP features in Java	L-13
4.	Write a Java program (using OOP features) to implement following scheduling algorithms: FCFS , SJF (Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive)	L-21
5.	Write a Java program to implement Banker's Algorithm	L-29
6.	Write a Java Program (using OOP features) to implement paging simulation using 1. Least Recently Used (LRU) 2. Optimal algorithm	L-33