Subject Name : System Software

Subject Code : CE 504

Teaching Scheme (Credits and Hours)

Teaching scheme				Total Credit	Evaluation Scheme					
L	T	P	Total		Theory		Mid Sem Exam	CIA	Pract.	Total
Hrs	Hrs	Hrs	Hrs		Hrs	Marks	Marks	Marks	Marks	Marks
03	00	02	05	04	3	70	30	20	30	150

Learning Objectives:

- To understand the relationship between system software and machine architecture.
- To understand the processing of an HLL program for execution on a computer.
- To understand the process of scanning and parsing.
- To know the design and implementation of assemblers, macro processor, linker and compiler.
- To have an understanding of loader, system software tools.
- To understand and know the working of device drivers

Outline of the Course:

Sr. No	Title of the Unit	Minimum Hours
1	Introduction to System Software and software tools	8
2	Assemblers	8
3	Macros and Macro Processors	9
4	Interpreters and Introduction of Compilers	10
5	Linkers and Loaders	10

Total hours (Theory): 45

Total hours (Lab): 30

Total hours: 75

Detailed Syllabus:

Sr. No	Topic	Lecture Hours	Weightage(%)
1	**Introduction to System Software and software tools :** **Language Processors:** • Introduction • Language Processing Activities • Fundamentals of Language Processing & Language Specification • Language Processor Development Tools. **Data Structures for Language Processing:** • Search Data structures • Allocation Data Structures. **Software Tools:** • Software Tools for Program Development • Editors • Debug Monitors • Programming Environments • User Interfaces.	8	18
2	**Assemblers:** • Elements of Assembly Language Programming • A Simple Assembly Scheme • Pass Structure of Assemblers • Design of a Two Pass Assembler • A single pass Assembler for IBM PC.	8	18
3	**Macros and Macro Processors:** • Macro Definition and Call • Macro Expansion • Nested Macro Calls • Advanced Macro Facilities • Design of a Macro Preprocessor.	9	20
4	**Interpreters and Introduction of Compilers:** • Interpreters: Use and overview of interpreters • Pure and impure interpreters. • Phases of the Compiler • Introduction of scanning and parsing • Aspects of compilation	10	22
5	**Linkers and Loaders:** • Introduction to linkers • Relocation and Linking Concepts • Design of a Linker • Self-Relocating Programs • A Linker for MS-DOS • Linking for Overlays and Loaders	10	22
	Total	**45**	**100**

Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lecture and laboratory which carries 10 marks in overall evaluation.
- One internal exam will be conducted as a part of internal theory evaluation.
- Assignments based on the course content will be given to the students for each unit and will be evaluated at regular interval evaluation.
- Surprise tests/Quizzes/Seminar/tutorial will be conducted having a share of five marks in the overall internal evaluation.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments shall be performed in the laboratory related to course contents.

Learning Outcome:

On successful completion of the course, the student will:

- Be able to compare various system software related to the given system
- Be able to understand the concepts required to develop the system software
- Be able to make proper use of system software tools

Text Book:

1. D. M. Dhamdhere, “Systems Programming and Operating Systems”, Second Revised Edition, Tata McGraw-Hill, 1999.

Reference books:

1. Leland L. Beck, “System Software – An Introduction to Systems Programming”, 3rd Edition, Pearson Education Asia, 2000.
2. Santanu Chattopadhyay, “System Software”, Prentice-Hall India, 2007
3. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, “Compilers: Principles, Techniques, and Tools”,2nd Edition, Pearson Education Asia

List of experiments:

Sr. No	Name of Experiment
1	Write a program to display lexemes from the given input file
2	Write a program to identify keywords and identifiers from the given input file
3	Write a program to insert, search and update the identifiers in the symbol table.
4	Write a program to insert, search and update the identifiers in symbol table using hashing.
5	Write a program to implement Recursive decent parser for given grammar
6	Implement pass-I of a two pass assembler.
7	Implement pass-II of a two pass assembler.
8	Implement pass-I of a macro processor and generate all the required tables.
9	Implement pass-II of a macro processor.
10	Implement absolute loader.