SARDAR RAJA COLLEGE OF ENGINEERING

RAJA NAGAR, ALANGULAM

DEPARTMENT OF COMPUTER APPLICATIONS

SUBJECT NAME : SYSTEM SOFTWARE

SUBJECT CODE : MC7203

DEPARTMENT : M.C.A

YEAR : I

SEMESTER : II

S.MAGESHWARAN

AP/ MCA

MC7203 SYSTEM SOFTWARE L T P C 3 0 0 3

COURSE OBJECTIVES

- To understand the relationship between system software and machine architecture, design and implementation of assemblers, linkers and loaders.
- To understand the design, function and implementation of assemblers, linkers and loaders
- To have an understanding of macro processors and system software tools.

UNIT I BASICS OF SYSTEM SOFTWARE AND ASSEMBLER 9

Introduction – System software and SIC/XE machine architecture - Basic assembler functions – Assembler algorithms and data structures – Machine dependent assembler features, Instruction formats and addressing modes – Program relocation – Machine independent assembler features – Literals – Symbol-defining statements – Expressions – Program Blocks – Control Sections and Program Linking-Implementation examples MASM assembler.

UNIT II COMPILER- LEXICAL ANALYSIS, SYNTAX ANALYSIS 9

Phases of compiler-Lexical Analysis: Role of a Lexical analyzer, input buffering, specification and recognition of tokens, Finite Automata, Designing a lexical analyzer generator, Pattern matching based on NFA's. Syntax Analysis: Role of Parser, Top-down parsing, recursive descent and predictive parsers (LL), Bottom-Up parsing, Operator precedence parsing, LR, SLR and LALR parsers.

UNIT III COMPILER- CODE GENERATION, OPTIMIZATION 9

Intermediate languages: graphical representations, DAGs, Three address code, types of three address statements, syntax directed translation into three address code, implementation of three address statements-Code Optimization: Machine dependent and machine independent code generation: Sources of optimization-Code Generation-Semantic stacks, evaluation of expressions, control structures, and procedure calls.

UNIT IV LOADERS AND LINKERS 9

Basic loader functions: Design of an Absolute Loader – A Simple Bootstrap Loader Machine dependent loader features Relocation – Program Linking – Algorithm and Data Structures for Linking Loader. Machine-independent loader features – Automatic Library Search – Loader Options Loader design options – Linkage Editors – Dynamic Linking – Bootstrap Loaders. Implementation examples: MSDOS linker.

UNIT V MACRO PROCESSORS & OTHER SYSTEM SOFTWARE 9

Basic macro processor functions – Macro Definition and Expansion – Macro Processor Algorithm and data structures – Implementation examples: MASM Macro Processor- Text editors – Overview of Editing Process - User Interface – Editor Structure – Interactive Debugging Systems – Debugging functions and capabilities – Relationships with Other parts of the system – User Interface Criteria - Virtual Machines.

TOTAL: 45 PERIODS

REFERENCES:

1. Leland Beck - “System Software – An Introduction to Systems Programming”, Third Edition, Pearson Education, Inc., 2008

2. A.V. Aho, R. Shethi and Ulman; Compilers - Principles, Techniques and Tools, Second Edition, Pearson Education, 2002.

3. D. M. Dhamdhere, "Systems Programming and Operating Systems", Tata McGraw Hill Company, Second Edition, 2009.

4. John J. Donovan, “Systems Programming”, Tata McGraw Hill Company, Second Edition, 2000.

5. V. Raghavan, “Principles of Compiler Design”, Tata McGrawHill Education Publishers, 2010.

6. Srimanta Pal, “Systems Programming “, Oxford University Press, 2011.

COURSE OUTCOMES:

- Able to trace the path of a source code to object code and to executable file
- To design the front end of the compiler-scanner, parser
- Understand and identify the relationship between system software and machine architecture
- Analyze the functions of assembler, compiler, linker, and loaders
- Know the design and implementation of loaders and linkers

MICRO LESSON PLAN

HOURS	LECTURE TOPICS	TEXT BOOK
UNIT I BASICS OF SYSTEM SOFTWARE AND ASSEMBLER		
1	Introduction – System software and SIC/XE machine architecture	R1
2	Basic assembler functions – Assembler algorithms and data structures	
3	Machine dependent assembler features, Instruction formats and addressing modes	
4	Program relocation – Machine independent assembler features	
5	Literals – Symbol-defining statements	
6	Expressions – Program Blocks	
7	Control Sections and Program Linking **(AV Class)**	
8 , 9	Implementation examples MASM assembler	
UNIT II COMPILER- LEXICAL ANALYSIS, SYNTAX ANALYSIS		
10	Phases of compiler-Lexical Analysis:	R1
11	Role of a Lexical analyzer, input buffering,	
12	Specification and recognition of tokens, **(AV Class)**	
13	Finite Automata, Designing a lexical analyzer generator,	
14	Pattern matching based on NFA's.	
15,16	Syntax Analysis: Role of Parser, Top-down parsing,	
17	Recursive descent and predictive parsers (LL), Bottom-Up parsing,	
18	Operator precedence parsing, LR, SLR and LALR parsers.	
UNIT III COMPILER- CODE GENERATION, OPTIMIZATION		
19	Intermediate languages: graphical representations, DAGs,	R1
20	Three address code, types of three address statements,	
21	Syntax directed translation into three address code,	
22	Implementation of three address statements	
23,24	Code Optimization: Machine dependent and machine independent code generation:	
25	Sources of optimization-Code Generation	
26	Semantic stacks, evaluation of expressions,	
27	Control structures, and procedure calls. **(AV Class)**	
UNIT IV LOADERS AND LINKERS		
28	Basic loader functions: Design of an Absolute Loader	R1
29,30	A Simple Bootstrap Loader Machine dependent loader features Relocation	
31	Program Linking – Algorithm and Data Structures for Linking Loader	
32	Machine-independent loader features – Automatic Library Search	
33	Loader Options Loader design options – Linkage Editors	
34	Dynamic Linking – Bootstrap Loaders **(AV Class)**	
35,36	Implementation examples: MSDOS linker	
UNIT V MACRO PROCESSORS & OTHER SYSTEM SOFTWARE		
37	Basic macro processor functions – Macro Definition and Expansion	R1
38	Macro Processor Algorithm and data structures	
39,40	Implementation examples: MASM Macro Processor	
40,41	Text editors – Overview of Editing Process **(AV Class)**	
42,43,44	User Interface – Editor Structure – Interactive Debugging Systems – Debugging functions and capabilities – Relationships with Other parts of the system –	
45	User Interface Criteria - Virtual Machines	