Faculty of Engineering and Technology

Bachelor of Technology

Effective from Academic Year: 2020-21

Branch Name:	Computer Engineering / Information Technology/Information and Communication Technology
Semester/Year:	Semester V / Third Year
Subject Title:	System Programming
Subject Code:	1ET1030503
Pre-requisite:	Basic idea regarding fundamentals of computers and basic knowledge of computer programming

Course Objective:

- To enlighten the design ideas of different system softwares like Assembler, Compiler, Interpreter, Linker, Loader and Macro preprocessor, Utility Programs such as Text Editor and Debugger.
- To familiarize and encourage the students to use various software tools for developing System programs.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Theory (Marks)**		**Practical (Marks)**		**Total (Marks)**
				University Assessment	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	
03	-	02	04	70	30	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weightage (%)**
1	**Introduction to System Software and Language Processing:** System Software Vs. Application Software, Different System Software, Overview of Assembler, Linker, Loader, Macro Processor, Text Editor, Debugger, Compiler, Interpreter, Operating System, Language processors, Language Processing Activities, Fundamentals of Language Processing, Fundamentals of Language Specification, Language Processor Development Tools: LEX and YACC	3	10
2	**Assemblers**: Need of Assembler, Types of Assemblers , Elements of Assembly Language Programming, Design specification of an Assembler, Single pass and Two pass translation of an Assembler, Design of a two pass Assembler	7	16
3	**Macros and Macro processor**: Macro , Advantages of using Macro , Macro Vs Procedure , Macro Definition and Call , Macro Expansion , Nested Macro Calls , Advanced Macro Facilities, Functions of Macro Preprocessor, Design of a Macro preprocessor, Design of a Macro Assembler	7	16
4	**Compilers and Interpreters**: Introduction to Compilers, Phases of a compiler, Aspects of compilation, Static and Dynamic Memory Allocation, Memory allocation in block structured languages, Compilation of Expressions: Operand descriptor, Register descriptor, Expression tree, Intermediate code for	7	16

	expression, Parameter passing mechanism, Code optimizing techniques, Local and global optimization, comparison of compilers and interpreters, Use of interpreter, Overview of Interpretation, Pure and impure interpreter		
5	**Scanning and Parsing**: The Scanning Process ,Regular Grammars and Regular Expression, Finite state automata, The role of a parser, parse tree, abstract syntax tree, Elimination of Left recursion, Left factoring, Top down parsing-Recursive descent parser, LL (1) parser, Bottom up parsing-Shift-reduce Parser, Operator Precedence Parsing, LR Parsers	8	17
6	**Linker and Loader**: Fundamentals of Relocation and Linking , Relocation and Linking algorithm, Self-relocating programs, Overlays, Basic Loader functions – Design of absolute loader, Direct Linking Loader, Compile-and-Go Loaders, General Loader Schemes, Relocating Loaders, Bootstrap Loader, Dynamic Loader	4	15
7	**System Tools: Editors and Debuggers:** Types of text editors, Editor Structure, Types of Errors, Debugging Procedures, Classification of Debuggers, Dynamic/Interactive Debugger – Interactive debugging systems - Debugging functions and capabilities – Relationship with other parts of the system – User-Interface Criteria.	3	10

Course Outcome:

After learning the course, the students will be able :

- To understand design, analyze and implementation of one pass, two pass or multi pass assembler, loader and linker, macro processors.
- To critique the features of modern editing /debugging tools.
- To use Lex tool used for generating lexical analyzer and YACC to implement various parser
- To write macros as and when required to increase readability and productivity.
- To apply optimization principles on given code

List of Text Books:

1. System Programming and Operating Systems, D M Dhamdhere, Tata McGrawhill Publication

List of Reference Books:

1. John J. Donovan “Systems Programming”, Tata McGraw-Hill Edition, 1972.
2. System Programming, Srimanta Pal, Oxford publication
3. Leland L. Beck, “System Software – An Introduction to Systems Programming”, 3rd Edition, Pearson Education Asia, 2000.
4. System Programming and Compiler Construction by R.K. Maurya & A. Godbole.
5. Aho. A.V. Sethi R. and Ulman J.D., Compilers, Principles, Techniques and tools, Addision-Wesley.

List of Suggested titles of Experiments :

1. Write a program to remove comment lines from C or CPP file.
2. Write a program to implement the lexical analyzer.
3. Implement a symbol table with functions to create, insert, modify, search, and display.
4. Write a program to develop expression evaluator.
5. Write a program to left factor the given grammar.

6. Write a program to remove the Left Recursion from a given grammar.
7. Write a program to implement DFA to check whether the given binary number is even.
8. Implement pass one and pass two of a two pass assembler.
9. Write a program which generates Quadruple Table for the given postfix String.
10. To study about LEX and YACC.
11. Implement following program using LEX.

    (i)Program to count the number of characters, words, spaces and Lines in a given input file. (ii)Program for counting number of vowels and consonant.

12. Implement using YACC a Program to recognize a valid arithmetic expression that uses operators +, -, * and /.