ITM(SLS)BarodaUniversity
SchoolofComputerScience,EngineeringandTechnology
B.Tech - Semester III

Course Name: System Software
Course Code: C2310C3
CourseType: Core

TeachingScheme :

TeachingScheme			Credits	ExaminationMarks				Total Marks
L	T	P	C	Theory Marks		Practicalmarks		
				External	Internal	External	Internal	
3	0	2	4	40	60	0	50	150

Preamble:

This course is concerned with those software tools and utilities which are an essential part of a Computer System which supports application software development. In order to apply a computer system to solve problems and provide computational services for a wide variety of human activities we need Application Software, e.g. an Accounting System package. However, to develop such an application package we need System Software like text Editors, Language processors - Compilers, Interpreters and Assemblers, several utilities like Linkers and program Loaders, and File Systems. Out of these the File System is generally studied as a part of Operating Systems, due to its close links with components of an Operating System and computer Hardware. The rest forms parts of the present course.

What is system Software?

Systems Programming is the activity of writing and maintaining system software.
System software is the layer between the hardware and application software; it controls the hardware, and provides services to applications.
The primary distinguishing characteristic of systems programming when compared to application programming is that application programming aims to produce software which provides services to the user directly (e.g. word processor),
Systems programming is the design, implementation, and maintenance of all of the programs like Operating System, assembler, linker, kernel, Bios, Macro Processor, Loader, Assembler etc.….

Prerequisite: Basis of Data Structures and Basic concepts of Operating System

Course Learning Outcome:
After completing the course, the student shall be able to:

	Course Outcome	**Bloom's Level**
CO1	Ability to understand Software and Its types and use text editors	Understanding
CO2	Ability to understand the Overview of Language Processors	Understanding
CO3	Ability to Understand the concept of Compiler	Understanding
CO4	Ability to Understand the Assembler& Implement Linker and Loaders	Applying
CO5	Ability to use and Implement Macro & Macro Processor	Application
CO6	Ability to understand Utility of Interpreter and Debuggers	Application

Course Syllabus:

Tasks #	***Topics (Programs) to be Completed***	***Total Hrs.***
1	**Overview of System Software and Text Editors:** Introduction, Software, Software Hierarchy, Systems Programming Tools, Life Cycle of a Source Program, Levels of System Software Text Editors: Overview of Editing Process, User Interface, Editor Structure, Text Editors-line-by-line (example ed), file-oriented WYSIWYG (example vi)	6
2	**Overview of Language Processors**: Programming Languages and Language Processors, Language Processing Activities, Program Execution, Fundamental of Language Processing, Symbol Tables Data Structures for Language Processing: Search Data structures, Allocation Data Structures.	6
3	**Assemblers** Elements of Assembly Language Programming, Design of the Assembler, Assembler Design Criteria, Types of Assemblers, Two-Pass Assemblers, One-Pass Assemblers **Macro and Macro Processors** Introduction, Macro Definition and Call, Macro Expansion, Nested Macro Calls, Advanced Macro Facilities, Design of a Macro Pre-processor, Design of a Macro Assembler, Functions of a Macro Processor, Basic Tasks of a Macro Processor, Design Issues of Macro Processors	10
4	**Macro and Macro Processors** Introduction, Macro Definition and Call, Macro Expansion, Nested Macro Calls, Advanced Macro Facilities, Design Of a Macro Pre- processor, Design of a Macro Assembler, Functions of a Macro Processor, Basic Tasks of a Macro Processor, Design Issues of Macro Processors, Features, Macro Processor Design Options, Two-Pass Macro Processors, One-Pass Macro Processors	10
5	**Scanning and Parsing** Programming Language Grammars, Classification of Grammar, Ambiguity in Grammatic Specification, Scanning, Parsing, Top Down Parsing, Bottom up Parsing, Language Processor Development Tools, LEX, YACC	

6	**Interpreter & Debuggers:** Benefits of Interpretation, Overview of Interpretation, The Java Language Environment, Java Virtual Machine, Types of Errors, Debugging Procedures, Classification of Debuggers, Dynamic/Interactive Debugger.	4
7	**Linker & Loader:** Introduction, Relocation of Linking Concept, Design of a Linker, Self-Relocating Programs, linking in Linux, Linking of Overlay Structured Programs, Dynamic Linking, Loaders, Different Loading Schemes, Sequential and Direct Loaders, Compile-and-Go Loaders	8
	Total	**42**

References:

Text Books:

Sr.No	**Book Name**
1	. "System Programming and Operating System", 2nd Ed, D.M.Dhamdhere, Tata-McGraw Hill
2	System Software –An Introduction to Systems Programming by Leland L. Beck, 3rd Edition, Pearson Education Asia, 2000

References:

Sr.No	**Book Name**
1	"Compilers - Principles and Practice", P.H.Dave and H.B.Dave, Pearson Education
2.	System Software by Santanu Chattopadhyay,Prentice-Hall India,2007
3.	System Programming by Srimanta Pal OXFORD Publication
4.	System Programming and Compiler Construction by R.K. Maurya & A. Godbole

Required Software:

1. GNU C Compiler: https://gcc.gnu.org/
2. For working with embedded system corresponding architecture tool chain : Assembler, Link editor, etc.

Web Resources:

1. https://texteditors.org/cgi-bin/wiki.pl
2. https://www.geeksforgeeks.org/editors-types-system-programming/
3. https://www.ques10.com/p/8931/write-a-brief-note-on-design-of-an-editor-2/
4. https://www.programming1011.com/2019/04/short-note-on-editors-in-system.html
5. http://www.tezu.ernet.in/~utpal/course_mat/ss_editor.html

Case Studies:

Sr. No	Case Studies	Evaluation
C1	Various compiler tools like GNU tool chain	1. Identification of algorithm. 2. Report preparation. 3. Presentation with VIVA
C2	Lexical Analyzer and YACC tools	

Related MOOCs courses

Sr. No	Course
M1	Introduction to System Software by Prof.S. Raman, Department of Computer Science and Engineering, IIT Madras. For More details on NPTEL visit http://nptel.iitm.ac.in

Other Videos Links:

Sr. No	About Video	Link
O1	Grammar and Its types	https://www.coursera.org/lecture/nand2tetris2/unit-4-3-grammars-rtIKX
O2	Syntax Analysis	https://www.coursera.org/lecture/nand2tetris2/unit-4-1-syntax-analysis-5pC2Z
O3	Lexical Analysis	https://www.coursera.org/lecture/nand2tetris2/unit-4-2-lexical-analysis-QM0lZ
04	Assembler construction by NPTEL-NOC IITM	https://www.youtube.com/watch?v=LEKDkhvUKjA
05	Front end loader by NPTEL IIT Guwahati	https://www.youtube.com/watch?v=DBTkILIZmSc

Lab Experiments:

S.No	Program Statement	Hrs
PS1	Write a C program to identify whether a given line is a comment or not	1
PS2	Write a C program to test whether a given identifier is valid or not.	1

PS3	Write a C program to simulate lexical analyzer for validating operators	1
PS4	Use macro features of C language	2
PS5	Write a program to left factor the given grammar	2
PS6		2
	Write a program to remove the Left Recursion from a given grammar	2
PS8	Write a program which generates Quadruple Table for the given postfix String	2
PS9	Write a C program to parse a given string using Predictive parsing for given grammar. Type→ simple | ↑id | array [simple] of type simple → integer | char | num dot dot num	2
PS10	Write a program to remove the Left Recursion from a given grammar	2
PS11	Write a Program to identify all the tokens from the source code	2
PSI2	C program to generate triple intermediate code for assignment statement.	2
	Total	**21**