|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Subject Code 20CST315** | **SYSTEM PROGRAMMING** | **L** | **T** | **P** | **S** | **C** |
| **Total Contact Hours : 45 Hours** | **3** | **0** | **0** | **0** | **3** |
|  | | | | | |
| **Pre-requisites:** | **Overview of System Programs Eg. Compiler, assembler, Linker, Loader and Operating system** | | | | | |
| **Co-requisites** | **Use of Editors and functions of system programs.** | | | | | |
| **Anti-Requisites** | **-** | | | | | |

**Course Objectives:**

* Design, write, and test moderately complicated low-level programs using a systems programming language.
* Proficiently use a preprocessor to implement code that is portable between different computing platforms.
* Implement routines that read and write structured binary files such as word processing documents, index systems, or serialized hierarchical data

**Course Outcomes:**

* To Understand different components of system software.
* To Understand the different phases and data structure used in assembly process by an assembler.
* To intermediate code generation in context of language designing.
* To Analyze the role of linkers and loaders in executing the programs
* To Recognize operating system functions such as memory management as pertaining to run time storage management.

**Contents of the Syllabus:**

**UNIT-I**

**Chapter-1 (Overview of System Software)[15h]**

Introduction, Software, Software Hierarchy, Systems Programming, Machine Structure, Interfaces, Address Space, Computer Languages, Tools, Life Cycle of a Source Program, Different Views on the Meaning of a Program, System Software Development, Recent Trends in Software Development, Levels of System Software.

**Chapter-2 (Assemblers)**

Elements of Assembly Language Programming, Design of the Assembler, Assembler Design Criteria, Types of Assemblers, Two-Pass Assemblers, One-Pass Assemblers, Single pass Assembler for Intel x86 , Algorithm of Single Pass Assembler, Multi-Pass Assemblers, Advanced Assembly Process, Variants of Assemblers Design of two pass assembler

**Chapter-3 (Macros)**

Macro and Macro Processors Introduction, Macro Definition and Call, Macro Expansion, Nested Macro Calls, Advanced Macro Facilities, Design Of a Macro Preprocessor, 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

**UNIT-II[15h]**

**Chapter -4(Compilers)**

Introduction to various translators, Various phases of compiler, Introduction to Grammars and finite automata, Bootstrapping for compilers, Lexical Analysis and syntax analysis, Intermediate Code Generation, Code optimization techniques, Code generation, Case study :LEXX and YACC, Design of a compiler in C++ as Prototype.

**Chapter -5(Scanning and Parsing)**

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

**Chapter -6(Debuggers)**

Introduction to various debugging techniques, Case Study: - Debugging in Turbo C++ IDE.

**UNIT-III[15h]**

**Chapter -7 (Linkers and Loaders)**

Introduction, Relocation of Linking Concept, Design of a Linker, SelfRelocating Programs, Linking in MSDOS, Linking of Overlay Structured Programs, Dynamic Linking, Loaders, Different Loading Schemes, Sequential and Direct Loaders, Compile-and-Go Loaders, General Loader Schemes, Absolute Loaders, Relocating Loaders, Practical Relocating Loaders, Linking Loaders, Relocating Linking Loaders, Linkers v/s Loaders

**Chapter -8 (Editors)**

Line editor, full screen editor and multi window editor, Case study MS-Word, DOS Editor and vi editor.

**Chapter – 9 (Operating System)**

Booting techniques and sub-routines, Design of kernel and various management for OS, Design of Shell and other utilities.

**ADVANCED TOPICS (BEYOND SYLLABUS)**

**Macro and Macro Processors, Scanning and Parsing**

## Text Books:

1. Donovan J.J., Systems Programming, New York, Mc-Graw Hill,1972.
2. Dhamdhere, D.M., Introduction to Systems Software, Tata Mc-Graw Hill1996.

## Reference Books:

* 1. Aho A.V. and J.D. Ullman Principles of compiler Design Addison Wesley/Narosa