Text Books

a) Introduction to Systems Software Dhamdhere, D.M. second Tata Mc-Graw Hill 1996

https://toaz.info/doc-view -Pdf link

b) Systems Programming Donovan J.J. 46/2009 Mc-Graw Hill 1972

c) Principles of compiler Design Aho A.V. and J.D. Ullman second Narosa Publishing House 2002

Lecturewise coverge:

Unit 1

Lecture 1: Overview of System Software: Introduction, Software, Software Hierarchy, Systems Programming

Lecture 2 : Overview of System Software: Machine Structure, Interfaces, Address Space, Computer Languages

Lecture 3 : Overview of System Software: Tools, Life Cycle of a Source Program

Lecture 4: Overview of System Software :Different Views on the Meaning of a Program, System Software Development

Lecture 5 : Overview of System Software: Recent Trends in Software Development, Levels of System Software

Lecture 6: Assemblers: Elements of Assembly Language Programming, Design of the Assembler, Assembler Design Criteria

Lecture 7: Assemblers: Types of Assemblers, Two-Pass Assemblers, One-Pass Assemblers

Lecture 8:Assemblers: Single pass Assembler for Intel x86 , Algorithm of Single Pass Assembler

Lecture 9:Assemblers :Multi-Pass Assemblers, Advanced Assembly Process

Lecture 10: Assemblers: Variants of Assemblers Design of two pass assembler

Lecture 11: Macros: Macro and Macro Processors Introduction, Macro Definition and Call, Macro Expansion

Lecture 12: Macros: Nested Macro Calls, Advanced Macro Facilities, Design Of a Macro Preprocessor

Lecture 13:Macros :Design of a Macro Assembler, Functions of a Macro Processor, Basic Tasks of a Macro Processor

Lecture 14: Macros :Design Issues of Macro Processors, Features, Macro

Processor Design Options

Lecture 15: Macros :Two-Pass Macro Processors, OnePass Macro Processors

Unit 2

Lecture 16: Overview of System Software: Doubt session

Lecture 17: Compilers: Introduction to various translators, Various phases of compiler

Lecture 18: Compilers: Introduction to Grammars and finite automata, Bootstrapping for compilers, Lexical Analysis

Lecture 19: Compilers: syntax analysis, Intermediate Code Generation

Lecture 20: Compilers: Code optimization techniques, Code generation

Lecture 21: Compilers: Case study :LEXX and YACC, Design of a compiler in C++ as Prototype

Lecture 22:Compilers: Classification of Grammar, Ambiguity in Grammatic Specification

Lecture 23: Compilers: Scanning, Parsing

Lecture 24: Scanning and Parsing: Top Down Parsing

Lecture 25: Scanning and Parsing: Bottom up Parsing

Lecture 26: Scanning and Parsing: Language Processor Development Tools and practice of grammer

Lecture 27: Scanning and Parsing: LEX, YACC

Lecture 28: Debuggers: Introduction to various debugging techniques

Lecture 29; Debuggers: Case Study: - Debugging in Turbo C++ IDE

Lecture 30: Debuggers: Doubt Session

Unit 3:

Lecture 31: Linkers and Loaders: Introduction, Relocation of Linking Concept, Design of a Linker

Lecture 32: Linkers and Loaders :Self Relocating Programs, Linking in MSDOS

Lecture 33: Linkers and Loaders :Linking of Overlay Structured Programs, Dynamic Linking

Lecture 34: Linkers and Loaders: Linking of Overlay Structured Programs,

Dynamic Linking
Lecture 35: Linkers and Loaders Loaders: Different Loading Schemes
Lecture 36: Linkers and Loaders: Absolute Loaders, Relocating Loaders, Practical Relocating Loaders
Lecture 37: Linkers and Loaders: Absolute Loaders:Relocating Loaders, Practical Relocating Loaders
Lecture 38: Linkers and Loaders: Linking Loaders, Relocating Linking Loaders, Linkers v/s Loaders
Lecture 39: Editors :Line editor, full screen editor and multi window editor
Lecture 40: Editors: Case study MS-Word
Lecture 41: Editors: DOS Editor
Lecture 42: Editors: vi editor
Lecture 43: Operating System: Booting techniques and sub-routines
Lecture 44: Operating System :Design of kernel and various management for OS
Lecture 45: Operating System: Design of Shell and other utilities.
ADVANCED TOPICS (BEYOND SYLLABUS) Macro and Macro Processors, Scanning and Parsing