

## System Programming

Semester V

Subject Code: BS51701

Lectures: 60

### Objectives:

The syllabus aims in equipping students with,

- The design and implementation issues of System programs that play an important role in program development
- Understanding the design structure of Assembler , macro processor and compilers
- The working of linkers and loaders and other development utilities.
- Introduction of Operating system as a software

### Unit 1: Introduction

5

#### Ch 1. Introduction

- Types of program – System program and Application program.
- Difference between system programming and application programming.
- Elements of Programming environment - Editor, Preprocessor, Assembler, Compiler, Interpreter, Linker and Loader, Debugger, Device drivers, Operating System.
- Simulation of a hypothetical machine -Memory, Registers, Condition Codes, Instruction format, Instruction Set, programs for hypothetical machine .

### Unit 2: Introduction to Editors and design structure of Assemblers, Macro preprocessor

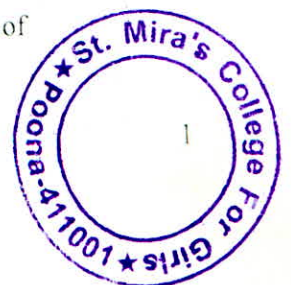
22

#### Ch 2. Editors

- Definition, need/purpose of editor.
- Types of editor- Examples ed, sed, VIM & emacs
- Structure of editor

#### Ch 3. Assembler

- Definition.
- Features of assembly language, advantages
- Statement format, types of statements – Imperative, Declarative, Assembler Directive.
- Constants and Literals.
- Advanced assembler directives (LTORG, ORIGIN, EQU).
- Design of assembler – Analysis Phase and Synthesis Phase.
- Overview of assembling process
- Pass Structure of Assembler – One pass, Two pass assembler.
- Problems of 1-pass assembler - forward reference, efficiency, Table of



Incomplete Instructions.

- Design of 2-pass Assembler – Pass-I and Pass-II
- Data structure of 2-pass assembler.
- Intermediate Code – Need, Forms-variant I and Variant II

#### **Ch 4. Macros and Macro Processors**

- Definition
- Macro definition and call
- Macro expansion – positional and keyword parameters
- Design of Data structures to be used for Macro definition and use
- Nested macro calls
- Advanced macro facilities – alteration of flow of control during expansion, expansion time variable, conditional expansion, expansion time loops. (with examples)
- Design of macro preprocessor – Design overview, data structure, processing of macro definition and macro expansion (Except algorithms).
- Macro assembler – Comparison of macro preprocessor and macro assembler. Pass structure of macro assembler

### **Unit 3: Compiler and Interpreter**

05

#### **Ch 5. Compiler and Interpreter**

- Definition, Aspects of compilation
- The structure of Compiler
- Phases of Compiler – Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code generation, code optimization,
- P-code compiler
- Use of interpreter, definition, Comparison with compiler,
- Overview of interpretation,
- Pure and impure interpreter.

### **Unit 4: Linker, Loader and Debugger**

08

#### **Ch 6. Linker and Loader**

- Introduction
- Concept of bindings, static and dynamic binding, translated, linked and load time addresses.
- Relocation and linking concept – program relocation, performing relocation, public and external references, linking, binary program, object module.
- Relocatability - nonrelocatable, relocatable, and self relocating programs (no algorithms), Linking for Overlays.
- Introduction to object file formats: a.out, ELF, COFF, EXE, PE and COM



**Unit 5: Introduction to Operating System as System Software**

08

**Ch 8. Introduction to Operating System**

- What Operating Systems Do – User View, System View, Defining OS
- Computer System Architecture – Single processor system, Multiprocessor systems, Clustered Systems
- Operating System Operations – Dual mode operation, Timer
- Process Management
- Memory Management
- Storage Management – File system management, Mass storage management, Caching, I/O systems
- Protection and Security
- Distributed Systems
- Special Purpose System – Real time embedded systems, Multimedia systems, Handheld systems,
- Computer Environment – Traditional computing, Client server computing, Peer to peer Computing

**\*Contact hours – 12 hours**

**Reference Books:**

**Reference Books:**

1. D.M.Dhamdhare, *Systems Programming and Operating Systems*, Second Revised Edition.[chapter 3,4]
2. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, *Compilers: Principles, Techniques, and Tools*
3. Leland L. Beck, *System Software - An introduction to Systems Programming*, Pearson Education [Chapter: 1].
4. John R. Levine, Elsevier Morgan Kaufmann, *Linkers and Loaders*, [chapter 6]  
Operating

