

# Course Code- CSC3501(Paper I)

## Operating System - I

### **Chapter -1**

### **Introduction**

# Course Outcomes (COs)

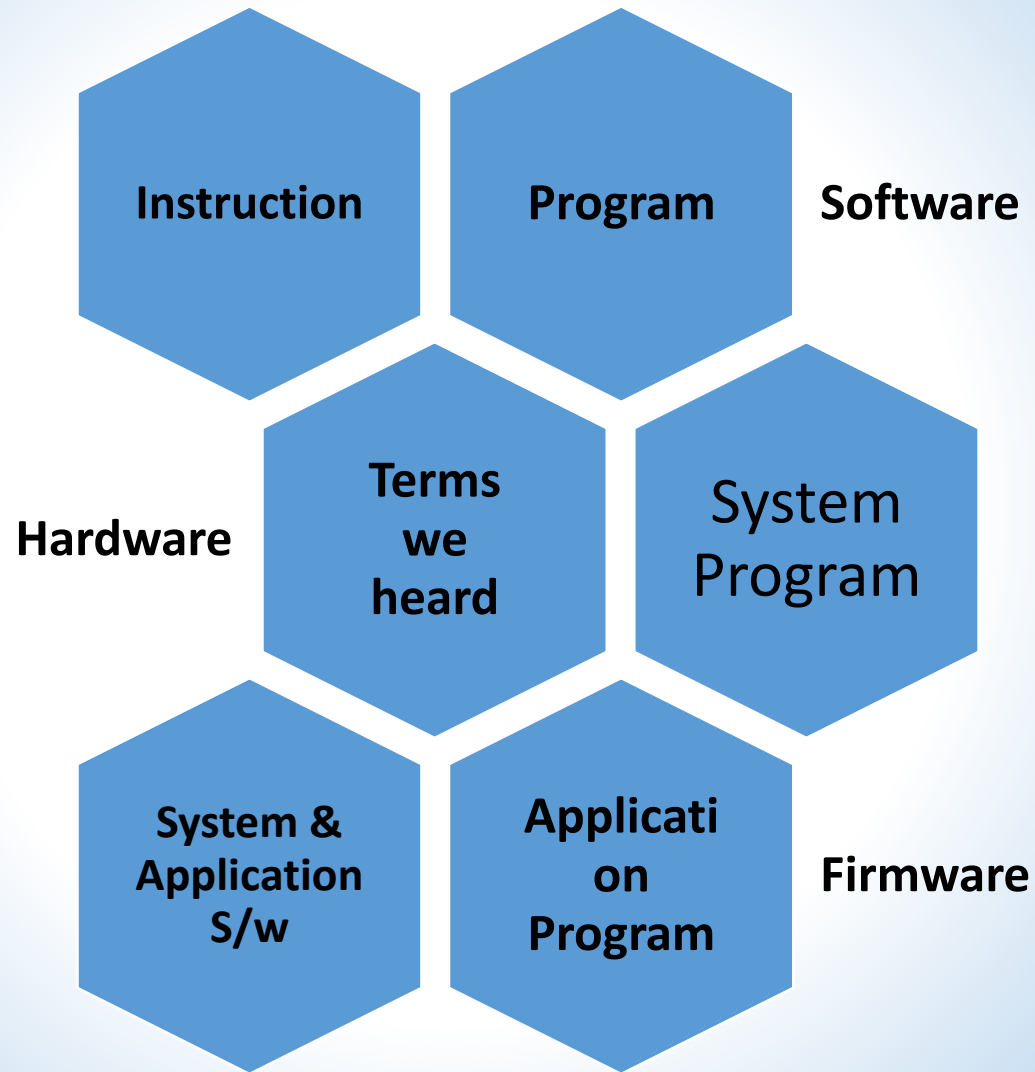
**On completion of the course, the students will be able to-**

CO1	List the elements of programming environment. Differentiate system programming and application programming
CO2	Write assembly programs and explain the process of translation, execution.
CO3	Discuss various system services and its examples, system calls and its types.
CO4	Apply the knowledge of process concept and Linux system calls to implement a command interface (Shell).
CO5	Compare computer system architectures. Discuss operating system operations
CO6	Transform the algorithms into code to implement the various system programs

# Contents

- **Types of Program-** System Program and Application Program
- **Differences between system programming and Application Programming**
- **Elements of programming environment-** Editor, Preprocessor, Assembler, Compiler, Interpreter, Linker and Loader, Debugger, Device Drivers, Operating System

# Introduction



# Computer Software System

## System Software

- A collection of system programs
- Bridges the gap between hardware, software and user interaction
- Translates user's need in the computer understandable language
- Machine dependent
- **System Program** : Each program in the system software
- **System Programming**: used to describe the techniques in designing system program
- Developed by system designer
- Example: Operating System

## Application Software

- A collection of application programs
- Are written for user specific tasks
- Machine independent
- **Application Program**: each program in application software
- **Application Programming**: used to provide a solution to specific problem using a computer tool
- Example: Ms-Office, Web Browser, Photoshop etc

**Activity 1: Write the differences between system programming and application programming**



# Elements of Programming Environment

- Assembler
- Loader
- Macros or Preprocessors
- Compilers
- Editors
- Debuggers
- Device Driver

# Editors



Fig. 1.1



# Assembler

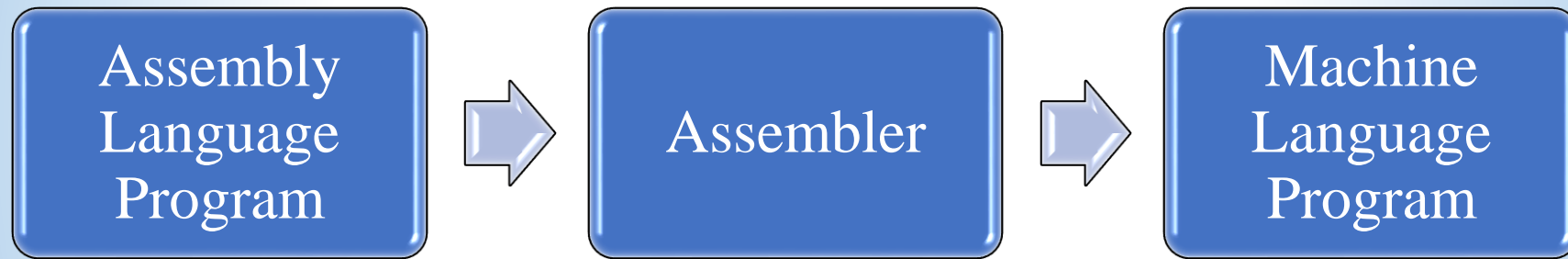


Fig: 1.2

# Macros



Fig. 1.3

# Compilers



Fig. 1.4

# Interpreters

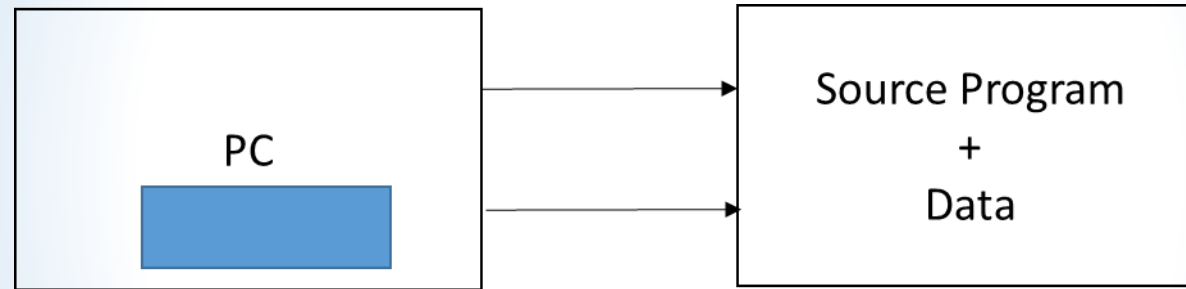


Fig. 1.5

# Linker



Fig. 1.6

# Loader



Fig. 1.7





# Debuggers

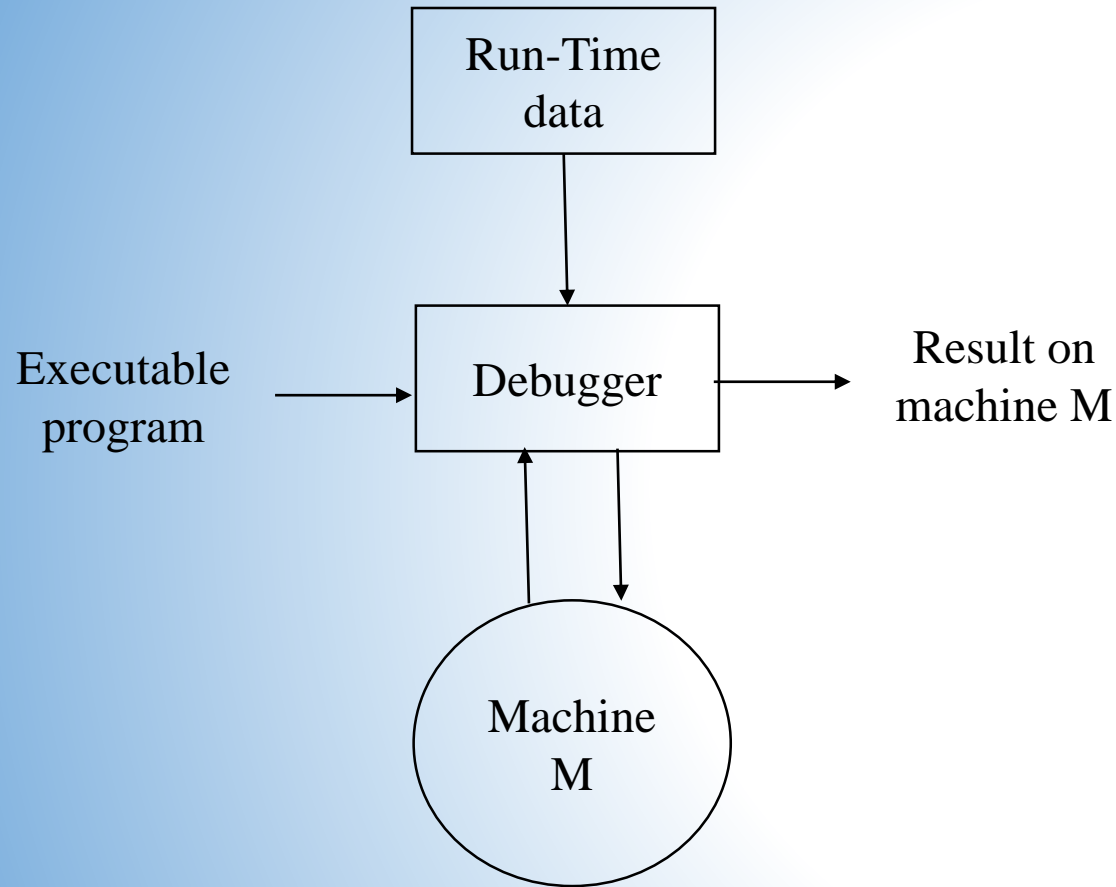


Fig. 1.8

- **Facilities :**

- a. setting breakpoints in the program
- b. Initiating a debug conversation when control reaches a breakpoint
- c. Displaying values of variables
- d. Assigning the values to the variables
- e. Testing user defined assertions and predicates involving program variables

**Debug Monitors:** used for Dynamic debugging

# Types of Debuggers

## Console Debugger

- GDB- GNU Debugger
- Most popular debugger for UNIX/GNU operating system
- Authored by “Richard M Stallman
- C,C++, Java, Assembly Programs
- Default interface – CLI
- Used in tracking Segmentation fault, Dump errors

## Visual Debugger

- Known as Data Display Debugger
- A GUI debugger
- One can visualize data using DDD
- No need to learn and remember GDB commands
- Allows to display structures and dereference pointers to create a visualization of a linked list or tree.

## **Device Drivers-**

- a system program which uses the functionality of the device when it is loaded in the memory.
- used to communicate with any device such as mouse, keyboard, printer etc.

## **Preprocessors-**

- used to enhance the ability of a given translator to include new features of high level language.

## **Operating System-**

- is a set of system program that act as an interface between user and the computer system

# Foundations of systems programming

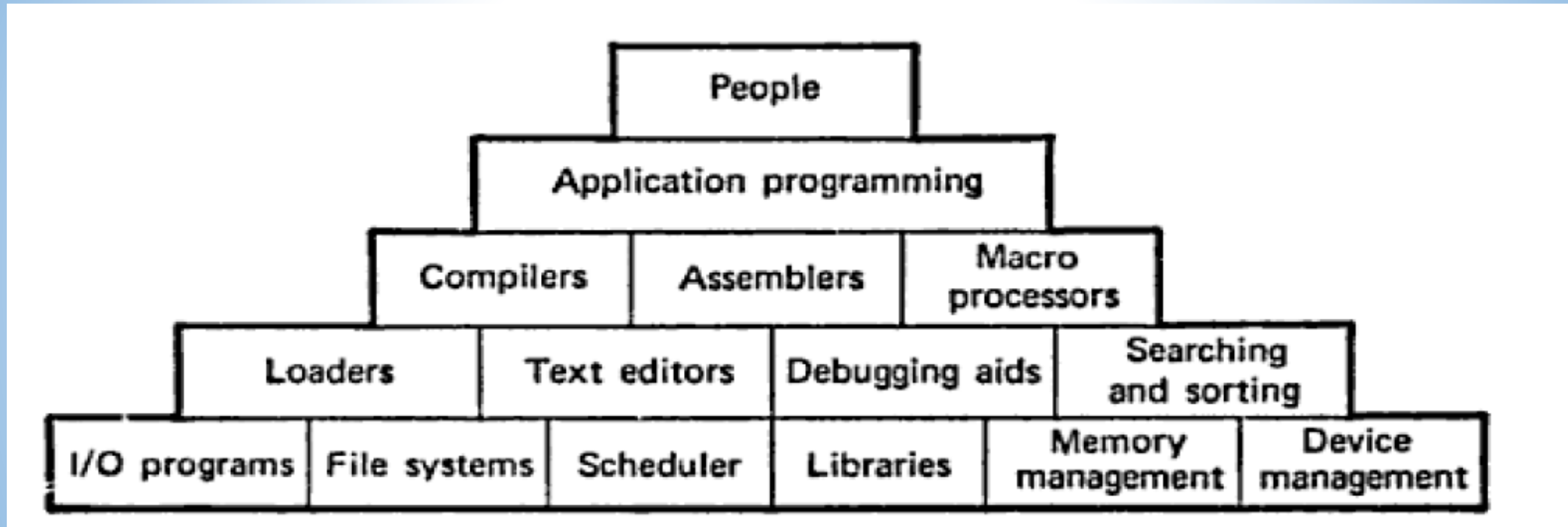


Fig. 1.9

Diagram Source: "Systems Programming", Author: John Donovan

## Multiple Choice Questions:

1. \_\_\_\_\_ software is used to load object code in main memory for execution.  
a) Linker                                      b) Assembler  
c) Macropreprocessor                      d) Loader
2. Which of the following translates the HLL source code into machine language line by line and execute it .  
a) Compiler                                    b) Assembler  
c) Editor                                        d) Interpreter
3. Name the element of programming language that replace each macro call with its macro-definition  
a) Macropreprocessor                      b) Assembler  
c) Editor                                        d) Compiler
4. Error correction can be done easily using \_\_\_\_\_  
a) Linker                                        b) Compiler  
c) Debugger                                    d) Interpreter
5. Unix environment supports two debuggers that are \_\_\_\_\_ & \_\_\_\_\_  
a) Turbo C and Borland                    b) gcc and cc  
c) Sdb and adb                                d) Ui and Text editor



# References:

## ❖ Books:

- D M Dhamdhere, Systems Programming , Tata McGraw Hill Education Private Limited
- John J Donovan, Systems Programming, Tata McGraw Hill

## ❖ Web references:

- <https://en.wikipedia.org/wiki/Debugger>
- <https://www.elprocus.com/what-is-debugging-types-techniques-in-embedded-systems/>
- [https://www.tutorialspoint.com/computer\\_programming/computer\\_programming\\_environment.htm](https://www.tutorialspoint.com/computer_programming/computer_programming_environment.htm)