

## Chapter-4

# SOFTWARE CONCEPTS

### ➤ Introduction

- A computer system has three components viz.
  - Hardware
  - Software
  - User
- **Hardware:** It consists of the physical components of a computer.
- **Software:** A set of instructions that tells the computer to perform an intended task.

### ➤ Types of Software

- Software is broadly classified into two categories namely,
  - System Software
  - Application Software

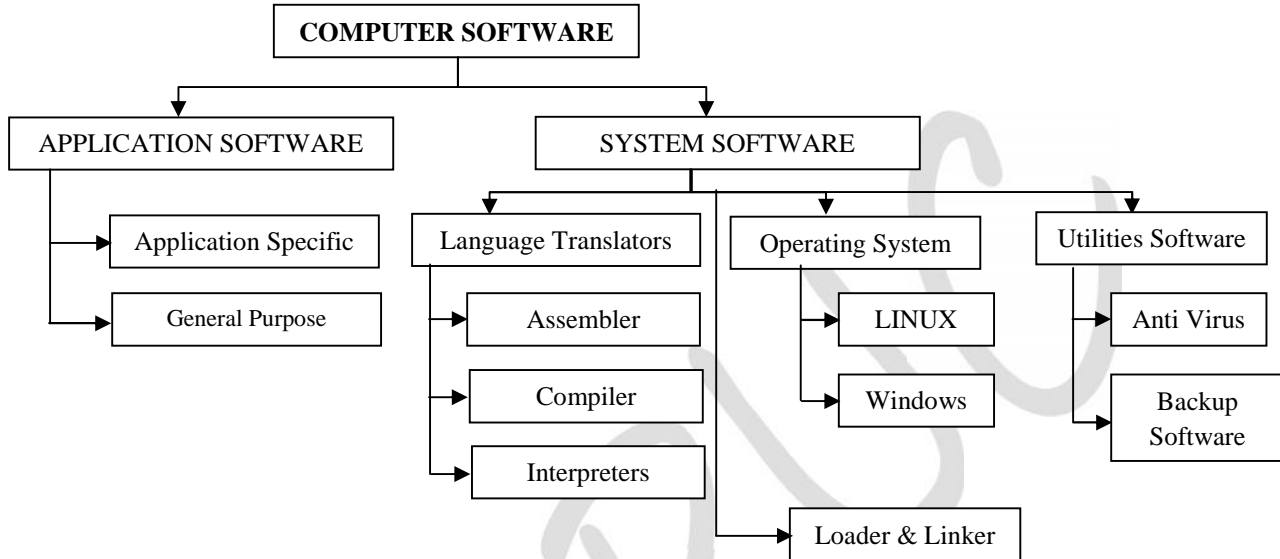
### ➤ System Software

- System software is a computer program that controls the system hardware and interacts with application software.
- System software is hardware dependent and not portable.
- System software provides a convenient environment for program development and execution.
- Programming languages like assembly language/C/C++/Visual C++/Pascal are used to develop the system software.
- System software is of three types:
  - Language Translators
  - Operating System
  - Utilities Software

### ➤ Application Software

- Application software that has been written to process performs a specific job.
- Application software is generally written in high level languages.
- Its focus is on the application, not the computing system.
- Application software is classified into two types:
  - Application Specific
  - General Purpose

- Application specific software is created to execute an exact task.
- It has a limited task. For example accounting software for maintaining accounts.
- General purpose software is not limited to only one function.
- For example: Microsoft office (MS-Word, MS-Excel), Tally, Oracle etc.



### ➤ Introduction to Computer Languages

- **Programming Language** is a set of rules called **syntax** which user has to follow, to instruct the computer what operation are to be performed.
- Computer language are classified into two categories:
  - Low-Level Languages
    - Machine level languages
    - Assembly languages
  - High-Level Languages
    - General Purpose languages (Ex: BASIC, PASCAL, C)
    - Specific purpose languages (Ex: COBOL, FORTAN, C++)

### ➤ Machine Level Language

- Machine level language is the fundamental language of a computer.
- It is written using binary numbers i.e. 0's and 1's.
- A program written in the machine level language is called **Machine code**.
- The instructions provided in machine language are directly understood by the computer and converted into electrical signals to run the computer.
- For example a typical program in machine language to add two numbers:

STATEMENTS	ACTION
0001 00110010	Load the data
0100 10100101	Add the contents
1000 00101001	Store the results
0000 00000000	Stop

- An instruction given in the machine language has two parts:
  - OPCODE (Operation Code)
  - Operand (Address/ Location)
- The first 4-bit represents Opcode denoting operation such as load, move, store etc.
- The last 8-bit represents the operand denoting the address.
- **Advantages:** It can be directly typed and executed and no compilation or translation is required.
- **Disadvantage:** These instructions are machine dependent and it is difficult to program, modify and debug errors.

### ➤ Assembly Level Language:

- Assembly level language is a low-level programming language that allows a user to write programs using letters, words and symbols called *mnemonics*, instead of the binary digits used in machine level languages.
- A program written in the assembly level language is called *Assembly code*.
- For example a typical program in assembly language to add two numbers:

STATEMENTS	ACTION
STA A	Load the data to accumulator
ADD B	Add the contents of B to Accumulator
STR C	Store the results in location C
PRT C	Print the results
HLT	Stop

- However a program in assembly language has to be converted to its equivalent machine language to be executed on computer.
- The translator program that converts an assembly code into machine code is called an **assembler**.
- **Advantages:** Mnemonic code are easy to remember, easy to understand, easy to modify and debug.
- **Disadvantage:** These languages are the mnemonic are machine dependent and assembly language programming takes longer to code.

### ➤ High-level Languages

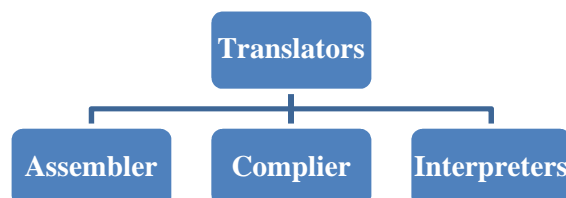
- A language designed to make programming easier through the use of familiar English words and symbols.
- High-level languages used English like language, which are easier to learn and use.
- High-level languages are machine independent. Therefore, a program written for one computer can be executed on different computers with no or only slight modifications.
- Some of the high-level languages are C, C++, JAVA, FORTRAN, QBASIC, and PASCAL.
- For example a typical program in high level language to add two numbers:

```
cin>>a>>b;
c = a + b;
cout<< " Answer = " << c;
```

- However a program in high-level language has to be converted to its equivalent machine language to be executed on computer.
- The translator program that converts an high level code into machine code is called an **compiler**.
- **Advantage:**
  - HLL's are machine independent.
  - Easy to learn and understand.
  - Easy to modify and debug the program.
- **Disadvantage:**
  - HLL is slower in execution.
  - HLL requires a compiler to convert source code to object code.
  - HLL take more time to execute and require more memory.

### ➤ Language Translators

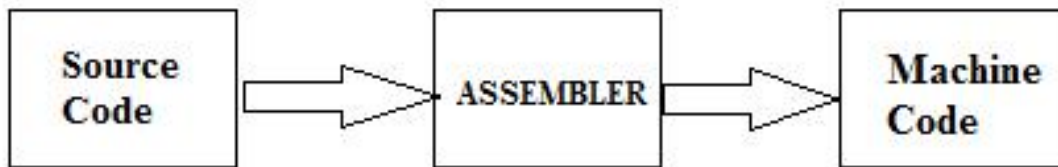
- The translator translates the high-level language to low level language.
- There are three types of translators: Compilers, Interpreters, and Assemblers.



### ➤ Assembler:

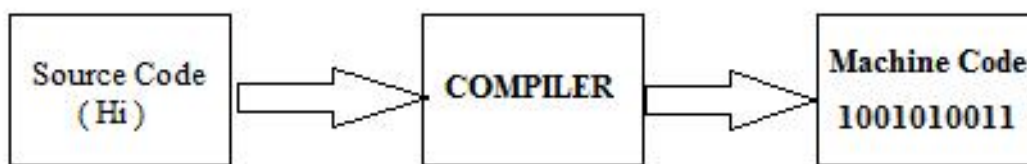
- Assembler is system software which translates an assembly language program into its machine language.

- It recognizes the mnemonics used in the assembly level languages and substitutes the required machine code for each instruction.
- Example: TASM (Turbo Assembler), MASM (Microsoft Macro Assembler) etc.



### ➤ **Compilers:**

- Compiler is system software that translates high-level language (source code) into the machine level language (machine/object code).
- It reads the whole program and translates the entire program at once into a series of machine level language instructions.
- Once compiled, the program normally gets saved automatically and can be executed directly.
- Examples: C, C++.



### ➤ **Interpreters:**

- An Interpreter reads once a statement of a high-level language program at a time and translates it into machine level language and executes it immediately.
- It continues to read, translate and execute the statements one by one until it reaches the end of the program.
- Therefore, it is slower than a compiler.
- The machine code produced by the interpreter is not saved and hence, to execute a statement again, it has to be interpreted again.
- Example: BASIC, PROLOG

### ➤ **Linker and Loader:**

- A source program written in high-level languages may contain a number of modules or segments. To execute properly the modules are to be linked so that execution of the program is sequential.
- This operation is performed by software called as the linker.
- **A linker is system software that links (combines) smaller programs to form a single program.**

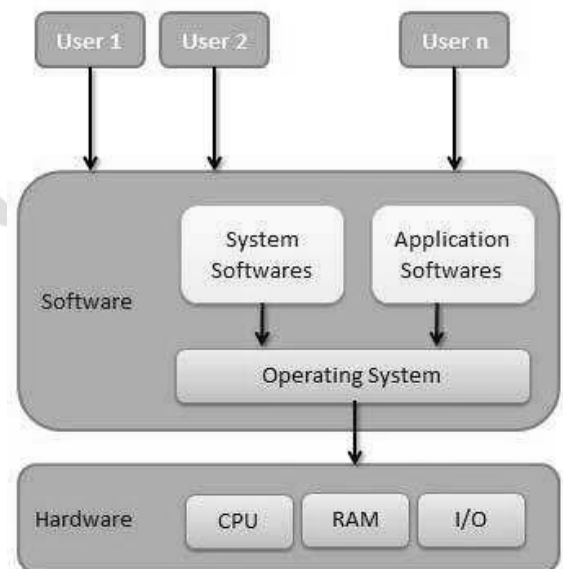
- Once an executable program is generated someone will have to load the program into the main memory of the computer so that it can be executed.
- This operation is performed by software called as the loader.
- **A loader is system software that loads machine code of a program into the system memory and prepares these programs for execution.**

### ➤ Utility Software

- Utilities are those helpful programs that assist the computer by performing helpful functions like backing up disk, scanning/cleaning viruses etc.
- Utility software is generally called as Application oriented ready-made system programs.
- Some of the important utility software is: Text editor, Backup utility, Disk Defragmenter, Antivirus software.

### ➤ Operating System(OS):

- An operating system is a set of programs which acts as an interface between the user and the computer.
- The main system software is the operating systems, which starts up the computer and control its operation.
- Operating system manages the resource, resolves conflicts' and tries to optimize the performance of the system.
- The user communicates with the computer through operating system commands and program instructions.
- Operating System plays an important role in loading programs from disk into memory, displaying message, translating program and in outputting the results.
- Some examples of operating system are: UNIX, LINUX, MS-DOS, and Microsoft Windows such as Win 95, Win 98, Win 2000, Win XP, Win 7, Win 8, Android, Mac OS etc.



### ➤ Functions of Operating System(OS):

- Main functions of the operating system are as follows:
  - Memory Management
  - Process Management
  - Device Management
  - File Management
  - Protection and Security
  - User interface or Command interpreter.

**✓ Memory Management:**

- The activities of memory management handled by OS are:
  - Allocate memory
  - Free Memory
  - Re-allocate memory to a program when a used block is freed
  - Keep track of memory usage.

**✓ Process Management:**

- The activities of process management handled by OS are:
  - Control access to shared resources like file, memory I/O and CPU
  - Control execution of applications
  - Create, execute and delete a process
  - Cancel or resume a process
  - Schedule a process
  - Synchronization, communication and deadlock handling for process

**✓ Device Management:**

- The activities of device management task handled by OS are:
  - Open, close and write device drivers.
  - Communication, control and monitor the device drivers.

**✓ File Management:**

- The activities of file management task handled by OS are:
  - Create and delete both files and directories
  - Provide access to files
  - Allocate space for files
  - Keep back-up of files
  - Secure files

**✓ Protection and Security:**

- OS protects the resource of system.
- User authentication, file attributes like read, write, encryption and back-up of data are used by OS to provide basic protection.

**✓ User Interface or Command Interpreter:**

- OS provides an interface between the user and the computer hardware.

- The user interface is a set of commands or a Graphical User Interface via which the user interacts with the application and the hardware.

### ➤ **Types of Operating System(OS):**

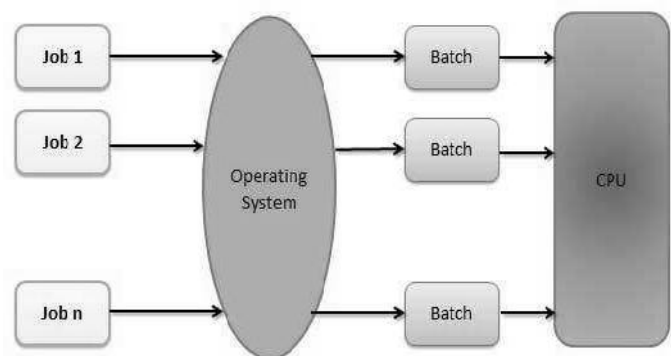
- The different types of operating system are:
  - Single user Operating System
  - Batch Operating System
  - Multiprogramming Operating System
    - Multitasking Operating System
    - Multiuser Operating System
    - Time sharing System (Online /Multiuser)
    - Real time system
  - Distributed Operating System
  - Network Operating System
  - Multithreading Operating System

### ✓ **Single User Operating System:**

- This OS allows only one user to share the system resource including the CPU.
- These are mainly the operating system configured for the use of desktop PC and laptops.
- DOS and Windows- 95, Win-98, Apple Macintosh etc are example.

### ✓ **Batch Operating System:**

- Batch is defined as a group of job with similar needs and similar resource requirements.
- The operating system that allows users to create batches and execute each batch sequentially, processing all jobs of batch considering them as a single process is called “Batch Operating system”



- It allows little or no interaction between users and executing programs.
- This is well suited for applications with large computation time and no user interaction.
- Payroll, forecasting, statistical analysis are programs example for its usage.

### • **Advantages:**

- User need not wait during its execution.



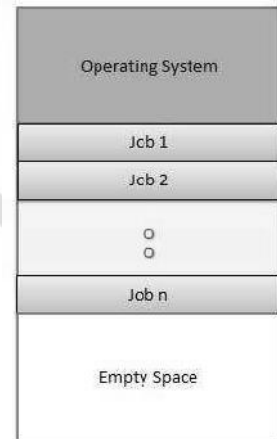
- It will function in FIFO (First In First Out) order.

- **Disadvantages:**

- Non-interactive mode of execution
- Offline debugging
- IBM System/360 Operating system is an example for Batch Operating system
- In DOS, we can emulate the batch processing using .BAT file.

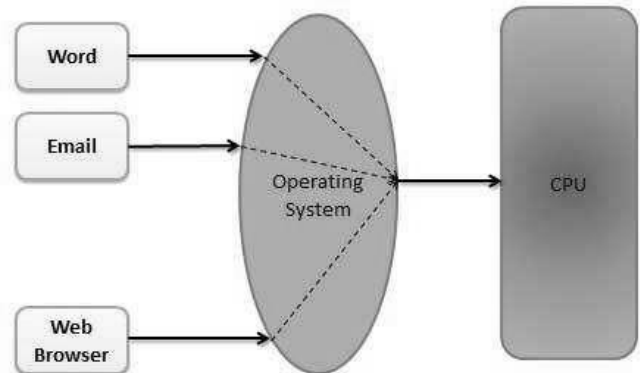
- ✓ **Multiprogramming Operating System:**

- Multiprogramming is the capability of CPU to execute two or more programs concurrently.
- Two or more programs are stored concurrently in primary storage, and the CPU moves from one program to another, partially executing each in turn.
- Early computer system and many personal computers execute program in the order in which it is read into the system.
- Only one program is executed at a time.



- ✓ **Multitasking Operating System:**

- Multitasking operating systems allow more than one program to run at a time.
- It gives you the perception of 2 or more tasks running at the same time. It does this by dividing system resources amongst these tasks.
- Multitasking is usually implemented by code and data of several programs in memory simultaneously and multiplexing processor and I/O device among them.
- Multitasking is also called context switching.
- Multitasking usually refers to a single user.
- Example: Windows 98

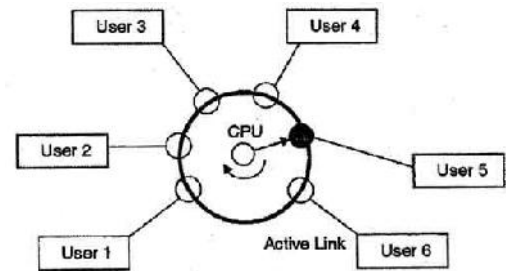


- ✓ **Multi-user Operating System:**

- A multi-user operating system allows multiple users on different computers or terminal to access a single system with one OS on it.
- It creates and maintains individual user environments, individual authentication and security level privileges, provides per user resource usage accounting.

### ✓ Time-Sharing Operating System:

- Time-Sharing operating system allows many users to share the computer resource simultaneously.
- Time sharing refers to the allocation of computer resource in time slots to several programs simultaneously.
- Large CAD and text processing systems belong to time-sharing OS.
- Most of the time-sharing operating systems adopted time-slicing/ round robin scheduling algorithm.
- Each user/process will receive a portion of the time slot.
- UNIX, Windows Server is the example for multiprogramming, multi-use and time sharing systems.



### ✓ Real-time Operating System:

- Real time systems refer to a computer and software systems that respond to events before the events become obsolete.
- The primary objective of real-time systems is to provide quick event – response times and thus meeting the scheduling deadlines.
- The main applications where real time systems are used include medical imaging systems, industrial control systems, flight control and military application.
- Windows CE, Symbian, LnuX OS are example for real-time systems.

### ✓ Distributed Operating System:

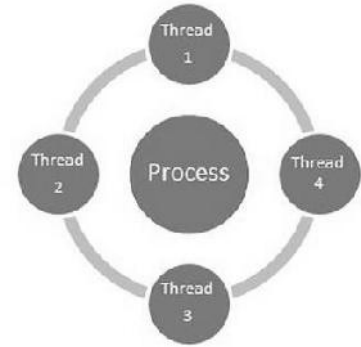
- A distributed operating system is a collection of autonomous computer systems capable of communication and cooperation via the software and hardware interconnections.
- For example: if we have 'N' systems in a distributed environment then the distributed OS helps us in balancing the load by sharing processors, I/O devices and memory etc.
- The ATM centers of a bank are example of distributed operating system.

### ✓ Network Operating System:

- A network operating system is a collection of software and associated protocols that allow a set of autonomous computer which are interconnected by a computer network.
- It can be used in a convenient and cost-effective manner.
- In a network operating system the user are aware of the existence of multiple computers and can log in to remote machines and copy files from one machine to another machine.
- Windows NT, windows server, LINUX are example.

✓ **Multithreaded Operating System:**

- A thread is a sequence of instructions within a program.
- A program or process may have many threads which share the same code section, data section, and other OS resources.
- It provides many logical paths through the program to be executed simultaneously.
- Sun Solaris, Windows 2000, Multithreaded UNIX, LINUX are examples.



➤ **Functional Features of Operating System(OS):**

- There are two types of user interface:
  - Command-Line User Interface (CUI)
  - Graphical User Interface (GUI)

✓ **CUI Interface:**

- CUI interface allows the user to interact with a computer through text terminal.
- It is a Non-graphical and text-based user interface.
- MS-DOS is an operating system which provides a CUI.
- The commands must be given at the command prompt ( C:\>)
- Both input and output are character based, the interface is also known as Character User Interface.

✓ **GUI Interface:**

- The GUI was first introduced in 1984 by Apple with Macintosh.
- GUI is an interface where the commands are selected from menus and icons rather the typing them from the keyboard.
- It allows user to click on the required icon, with the help of Mouse.
- Windows Operating System is the most popular OS based on GUI.

✓ **Difference between CUI and GUI:**

	CUI	GUI
1.	Command-line User Interface	Graphical User Interface
2.	The user must type the commands at command prompt to interact with the computer	The user must click on icons, menus, dialog boxes etc. to interact with the computer.
3.	The user must remember the commands	The user need not remember any

	and their parameters	commands as it is available in the form of menus and icons on monitor.
4.	Mouse operation not available for DOS	Mouse operation required to select the commands.

CHAPTER 4 – SOFTWARE CONCEPTS BLUE PRINT				
VSA (1 marks)	SA (2 marks)	LA (3 Marks)	Essay (5 Marks)	Total
-	01 Question	01 Question	-	05 Marks

\*\*\*\*\*