

Computer Fundamentals

Unit-IV

Mrs. Kiran Bala Dubey

Assistant Professor

Department of Computer Science

Govt. N. P. G. College of Science,

Raipur

Contents

- **System Software**

- System s/w Vs Application s/w
- Type of system s/w
- Introduction & Types of OS
- Boot loader
- Diagnostic programs
- BIOS
- Utility programs

- **Application Software**

- Microcomputer s/w
- Interacting with the system
- Trends in PC s/w
- Type of Application s/w
- Differences between program & packages

Languages

Definition
Generations of computer languages
Type of languages
Language processors : Assembler, Interpreter & Compiler
Linker & Loader
Program constructs
Algorithm & Flowchart

System Software

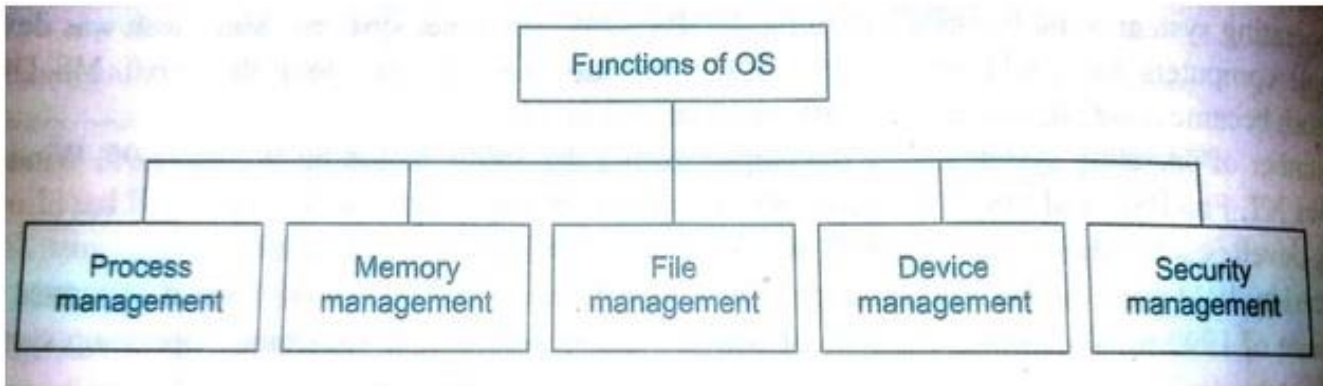
- System software is a set of one or more programs designed to control the operation and extend the processing capability of a computer system.
- System software makes the operation of a computer system more effective and efficient.
- There are following types of system software –
 - Operating system
 - Programming Language Translators
 - Communication Software
 - Utility Programs

Types of system software -

- **Operating system** – It takes care of effective and efficient utilization of all hardware and software components of a computer system.
Example - DOS, WINDOWS, UNIX.
- **Programming language translators** – It transform the instructions prepared by programmers in a programming language into a form that can be interpreted and executed by a computer system.
Example- Assembler, Compiler & Interpreter
- **Communication Software** – In a network environment communications software enables transfer of data and programs from one computer system to another.
- **Utility Programs**- It is a set of programs that help user in system maintenance tasks and in performing tasks of routine nature.
Example- Format, Backup, Restore, Zip, Unzip & Anti virus

Introduction & Types of OS

- An Operating System is the interface between the computer hardware and the end-user. Processing of data, running applications, file management and handling the memory is all managed by the computer OS.



Types of OS –

1. Batch Operating System

- There is no direct communication between the computer and the OS.
- There is an intermediate, the Operator, which needs to distribute the work into batches and sort similar jobs.
- Multiple users can use it.
- Can easily manager a large amount of work.

2. Real-Time Operating System

- It has a data processing system.
- The processing time is very small between the user's command and the output.
- Used in fields where the response needs to be quick and rapid.

3. Time-Sharing Operating System

- Multiple people at various terminals can use a program at the same time.
- The main motive is to minimize the response time.

4. Distributed Operating System

- When two or more systems are connected to each other and one can open files which are not present in their system but in other devices connected in the network.
- Its usage has now increased over the years.
- They use multiple central processors to serve real-time applications.
- Failure of one system does not affect the other systems connected in the network.

5. Embedded Operating System

- These special Operating systems are built into larger systems
- They generally are limited to single specific functions like an ATM.

6. Network Operating System

- They have one main server which is connected to other client servers
- All the management of files, processing of data, access to sharing files, etc. are performed over this small network
- It is also a secure operating system for working with multiple users.

7. Mobile Operating System

- With the advancement in the field of technology, smart phones now are released with an Operating system.
- They are designed in a manner that they can help a small device work efficiently.

8. Multi user operating systems

- It allows a number of users to work simultaneously on a single computer system. These types of operating systems are specially designed for the multi user system.
- Examples of multi user operating systems includes Unix, Linux and Windows 2000.

9. Multitasking operating systems

- It allows a user to give multitask at a same time on a single computer system.
- multitasking operating system are also known as multiprocessing operating system and multiprogramming operating system.
- The number of tasks processed simultaneously by this operating system is depending upon speed of CPU, the capacity of memory and size of programs.
- Examples of multitasking operating systems includes Unix, Linux, Windows 2000, windows XP and windows 10.

10. Multiprocessor operating system

- It allows the computer system to use more than one CPU in a single system for executing more than one or multiple processes at a time.
- A computer system having multiple CPU process faster than a system which contains a single CPU.
- Examples of multiprocessor operating systems are Linux, Unix, windows 2000 etc.

Boot loader

- A boot loader is a special operating system software that loads and starts the boot time tasks and processes of an operating system.
- It enables loading the operating system within the computer memory when a computer is started or booted up.
- A boot loader is also known as a boot manager or bootstrap loader.

Diagnostic programs

- A diagnostic program is a software tool used to diagnose problems with a particular set of hardware devices. It can be used by a trained technician or by the owner of the device, to identify and resolve hardware issues.
- Diagnostic programs ideally provide the user with guidance regarding any issues or problems found during its operation.

BIOS

- BIOS (basic input/output system) is the program in a ROM chip found on motherboards that uses to start the computer system after it is powered on. It also manages data flow between the computer's operating system (OS) and attached devices, such as the hard disk, video adapter, keyboard, mouse and printer.
- The main use of BIOS is to act as a middleman between OS and the hardware they run on.

Main functions of a BIOS are –

- **POST** - Test the computer hardware and make sure no errors exist before loading the operating system.
- **Bootstrap Loader** - Locate the operating system. If a capable operating system is located, the BIOS will pass control to it.
- **BIOS drivers** - Low-level drivers that give the computer basic operational control over your computer's hardware.
- **BIOS setup** or **CMOS setup** - Configuration program that allows you to configure hardware settings including system settings, such as computer passwords, time, and date.

Application Software

Application software is a set of one or more programs designed to solve a specific problem or task. Example Payroll processing system, Exam result processing system, Ticket reservation Software, word processor, a spreadsheet, an accounting application, a web browser, an email client, a media player, a file viewer, a console game, or a photo editor. etc.

Types of application software –

- Pre-written Application Software
- User specific Application Software

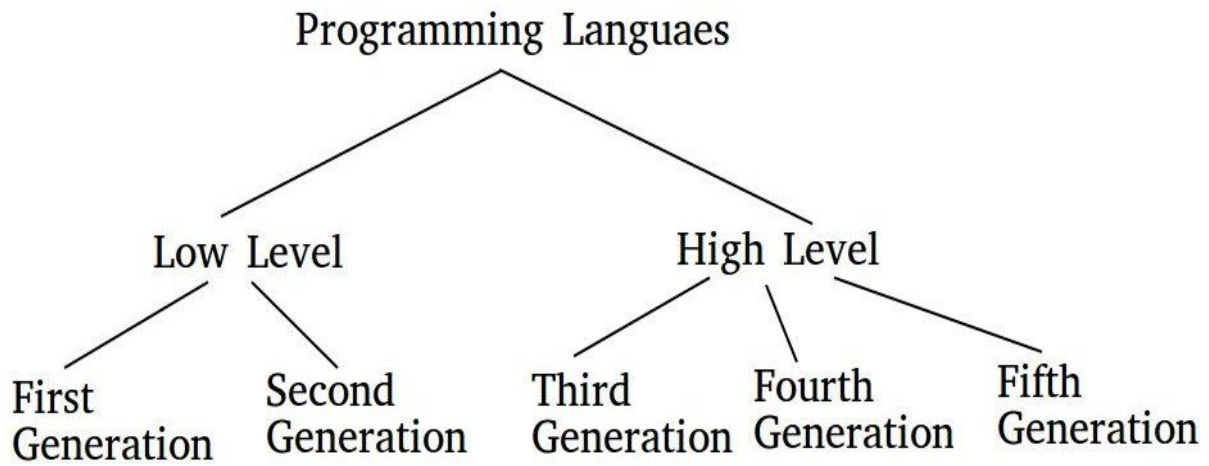
Differences between program & packages

1. Programs are set of instructions to perform a task.
2. These are one piece of software.
3. Programs are something that provide instruction to computer to perform particular task.
4. Program is especially developed by highly skilled programmer i.e. single programmer.
5. Programs are easy to develop and requires less time as compared to packages i.e. few hours or minutes.

1. Packages are set of software programs.
2. These are multiple pieces of software.
3. Packages are something that is packed.
4. It is a special method of distributing and installing software to computer.
5. Package is especially developed by group of programmers i.e. more than one programmer and require more time to develop than program i.e. weeks or months.

Differences between System Software & Application Software

Languages



- The **first generation languages**, or **1GL**, are low-level languages that are machine language.
- The **second-generation languages**, or **2GL**, are also low-level assembly languages. They are sometimes used in kernels and hardware drives, but more commonly used for video editing and video games.
- The **third-generation languages**, or **3GL**, are high-level languages, such as C, C++, Java, JavaScript, and Visual Basic.
- The **fourth-generation languages**, or **4GL**, are languages that consist of statements similar to statements in a human language. Fourth generation languages are commonly used in database programming and scripts examples include Perl, PHP, Python and SQL.
- The **fifth-generation languages**, or **5GL**, are programming languages that contain visual tools to help to develop a program. Examples includes Mercury, OPS5, and Prolog.

Language processors

The language processor is a special translator program used to translate the program written in high-level language (Source code) into machine code (Object code).

- **Assembler**- The Assembler is used to translate the program written in Assembly language into machine code.
- **Interpreter** - The translation of single statement of source program into machine code is done by language processor and executes it immediately before moving on to the next line is called an interpreter. If there is an error in the statement, the interpreter terminates its translating process at that statement and displays an error message. The interpreter moves on to the next line for execution only after removal of the error.
- **Compiler** - The language processor that reads the complete source program written in high level language as a whole in one go and translates it into machine language is called as a Compiler.

Linker & Loader

In execution of the program, major role is played by two utility programs known as **Linker** and **Loader**.

1. Linker :

A linker is special program that combines the object files, generated by compiler/assembler, and other pieces of codes to originate an executable file have. exe extension. In the object file, linker searches and append all libraries needed for execution of file. It regulates memory space that code from each module will hold. It also merges two or more separate object programs and establishes link among them.

2. Loader :

The loader is special program that takes input of object code from linker, loads it to main memory, and prepares this code for execution by computer. Loader allocates memory space to program. Even it settles down symbolic reference between objects. It in charge of loading programs and libraries in operating system.

Program constructs

- **Programs** are designed using common building blocks. These building blocks, known as programming constructs (or programming concepts), form the basis for all programs.
- There are three basic building blocks to consider:-
 - **sequence** is the order in which **instructions** occur and are processed
 - **selection** determines which path a program takes when it is running
 - **iteration** is the repeated **execution** of a section of code when a program is running
 - There are two types of iteration:
 - **definite iteration** (also known as count-controlled iteration)
 - **indefinite iteration** (also known as condition-controlled iteration)
- All programs use one or more of these constructs. The longer and more complex the program, the more these constructs will be used repeatedly.

Algorithm & Flowchart

Algorithm refers to a set of rules/instructions that step-by-step define how a work is to be executed upon in order to get the expected results.

In programming, algorithm is a set of well defined instructions in sequence to solve the problem.

Qualities of a algorithm

- Input and output should be defined precisely.
- Each steps in algorithm should be clear and unambiguous.
- Algorithm should be most effective among many different ways to solve a problem.
- An algorithm shouldn't have computer code. Instead, the algorithm should be written in such a way that, it can be used in similar programming languages.

Examples Of Algorithm

Write an algorithm to add two numbers entered by user.

Step 1: Start

Step 2: Declare variables num1, num2 and sum.

Step 3: Read values num1 and num2.

Step 4: Add num1 and num2 and assign the result to sum.






sum \leftarrow num1+num2

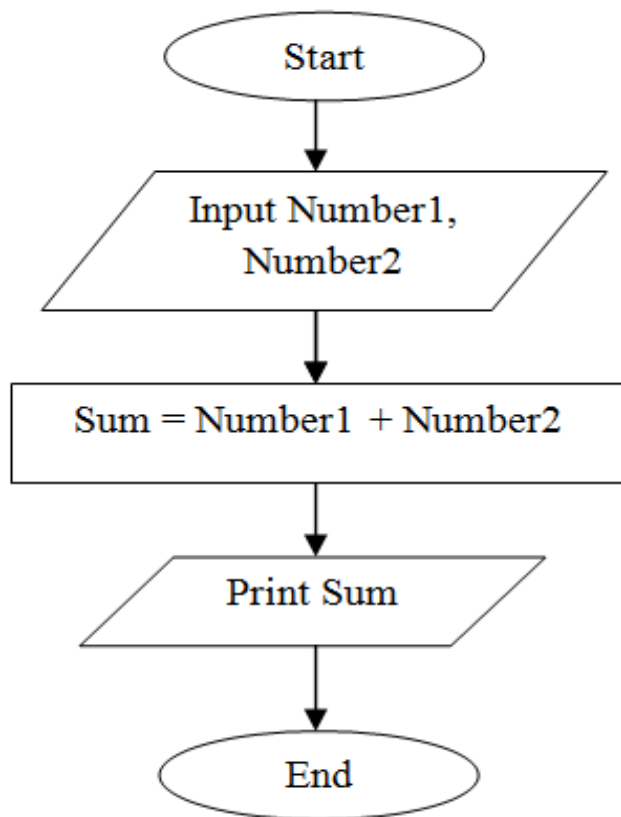
Step 5: Display sum

Step 6: Stop

Flowchart

- A flowchart is a type of diagram that represents an algorithm, workflow or process.
- The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.
- Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision



Algorithm

1. Algorithm is step by step procedure to solve the problem.
2. Algorithm is complex to understand.
3. In algorithm plain text are used.
4. Algorithm is easy to debug.
5. Algorithm is difficult to construct.
6. Algorithm does not follow any rules.
7. Algorithm is the pseudo code for the program.

Flowchart

1. Flowchart is a diagram created by different shapes to show the flow of data.
2. Flowchart is easy to understand.
3. In flowchart, symbols/shapes are used.
4. Flowchart it is hard to debug.
5. Flowchart is simple to construct.
6. Flowchart follows rules to be constructed.
7. Flowchart is just graphical representation of that logic.