Chapter IV

Software

Computer software, or just **software**, is any set of machine-readable instructions (most often in the form of a computer program) that directs a computer's processor to perform specific operations. The term is used to contrast with computer hardware, the physical objects (processor and related devices) that carry out the instructions. Hardware and software require each other and neither has any value without the other.

Note:

1. Firmware is software that has been permanently stored in hardware (specifically in non-volatile memory). Thus, it has qualities of both software and hardware.
2. Software is a general term. It can refer to all computer instructions in general, or to any specific set of computer instructions. It is inclusive of both machine instructions (the binary code that the processor understands) and source code (more human-understandable instructions that must be rendered into machine code by compilers or interpreters before being executed).

Program: -

Is a set of sequence instruction that tell the computer what to do?

Programmer: -

Is the person who makes the program using one of the computer's programming languages.

Software: -

Is a collection of programs, data, and information which tell the computer what to do & how to do it.

Type of software according to usage: -

1. System software. Is the basic software needed for a computer to operate (most notably the operating system);

2. Application software. Is all the software that uses the computer system to perform useful work beyond the operation of the computer itself?

Fig: Software

1. System software (operation system): -

Types of programs that hold all instruction that make the entire computer and loaded automatically when start the computer. Or software is a set of *programs*, *procedures*, *algorithms* and its *documentation* concerned with the operation of a data processing system.

e.g.: **MS – DOS, Window (PC).**

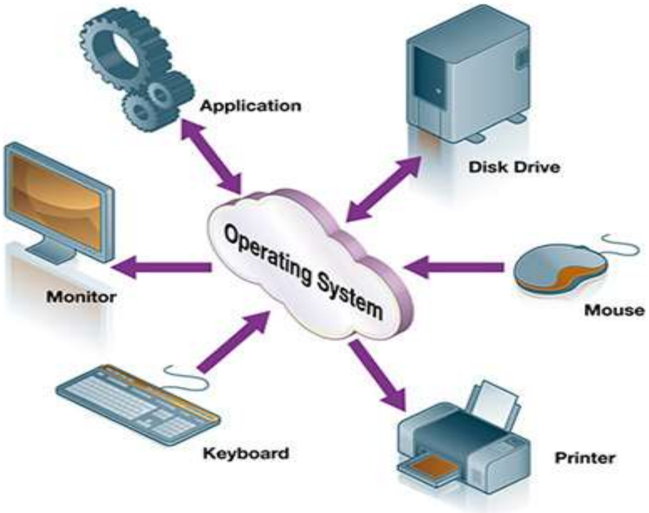


Fig: operating System.

a. Programming languages: - A set of words, rules, used by programmer to solve specific kind of problems.

e.g.: **FORTRAN, PASCAL, C^{++},** and **JAVA.**

Generation of programming languages: -

1. Machine language: -

It is written in (**1, 0)**, it is difficult language, which does not need any translator.

2. Assembly language: -

Deferred on using some of the mnemonic (brief) symbols, {like: MUL for multiply}, it uses addressing symbols.

Easier than machine language, but it still depends on the computer architecture, a translator is needed and is called ***Assembler***.

3. Third generation languages (or high level languages): -

Use statements that one can understand, it needs translator, it can be used very easily to solve complex problems.

e.g.: **Basic, C**, and **Pascal**.

4. Application Generations: -

Called fourth Generation languages, they are data base languages used to create files, forms, reports, without the written of any programs. It uses very simple statements or instruction; it does the job by telling the computer what the user needs, but without telling it how to do it.

These languages like: **Oracle, Access.**

5. Object oriented Language: -

These languages consist of a collection of objects; objects interact with each other by sending messages. An important characteristic of these languages is *Encapsulation.*

Encapsulation: -

Mean that the data & its operation are collected in the same place. You cannot access any data on any object without using the specific instruction for the object.

b- compiler & interpreter

Programs that translate the source code into the object code.

Compilers: -

Translates a whole program, the object code can be executed then.

Interpreter: -

Translates and executes one instruction at a time.

c. Operating system: -

The most important system software, it holds all the instruction that make the computer work.

1. Starting computer.
2. Displaying on the monitor.
3. Saving & loading files.
4. Using secondary memory.
5. Managing application software by being an interface between them and the computer’s equipment.

Functions of operating system: -

1. Booting up.
2. User interface.
3. Task & resources management.
4. Monitoring.
5. Files management.
6. System security.

Types of operating system: -

1. Multi – tasking OS.

In computing, multitasking is a method where multiple tasks, also known as processes, are performed during the same period of time. The tasks share common processing resources, such as a CPU and main memory.

In the case of a computer with a single CPU, only one task is said to be running at any point in time, meaning that the CPU is actively executing instructions for that task. Multitasking solves the problem by scheduling which task may be the one running at any given time, and when another waiting task gets a turn. The act of reassigning a CPU from one task to another one is called a context switch. When context switches occur frequently enough the illusion of parallelism is achieved.

Even on computers with more than one CPU (called multiprocessor machines), multitasking allows many more tasks to be run than there are CPUs. The term "multitasking" has become an international term, as the same word is used in many other languages such as German, Italian, Dutch, Danish and Norwegian.

Operating systems may adopt one of many different scheduling strategies, which generally fall into the following categories.

Note: OS mean operating system.

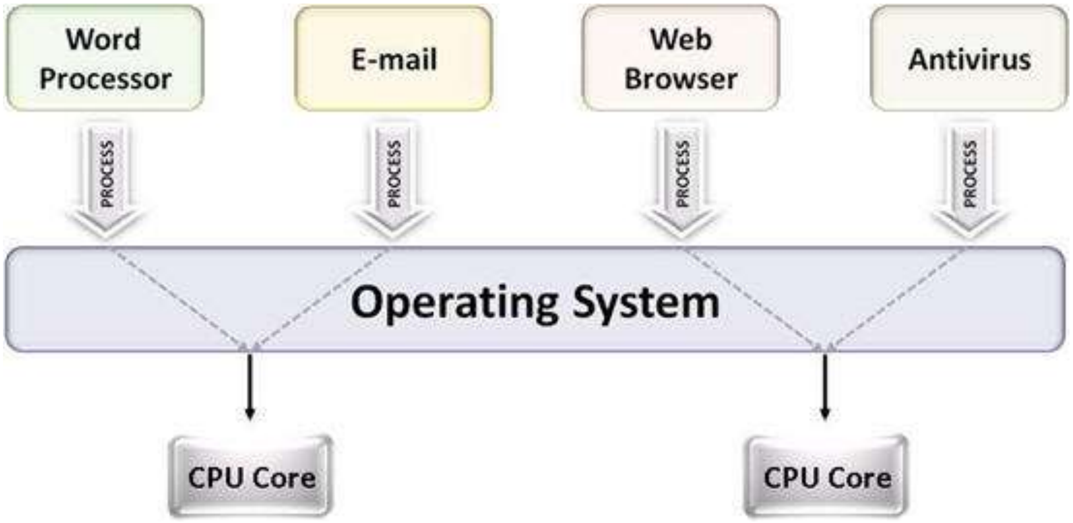


Fig: Multi-Tasking O S.

2. Multi – processing OS.
   Is the use of two or more central processing units (CPUs) within a single computer system? The term also refers to the ability of a system to support more than one processor and/or the ability to allocate tasks between them.

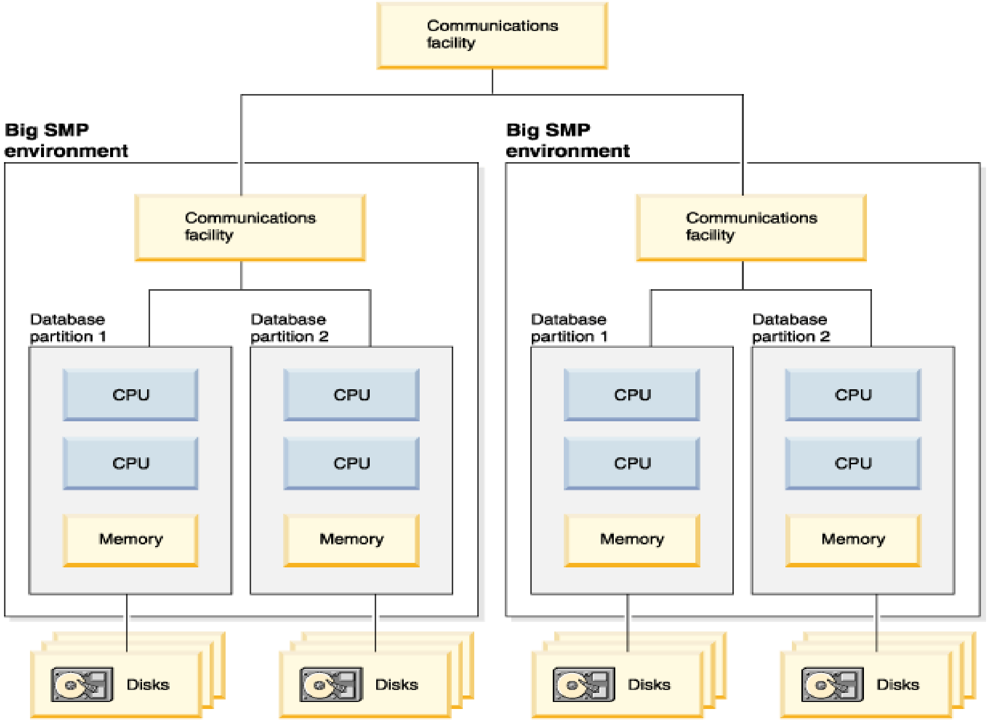


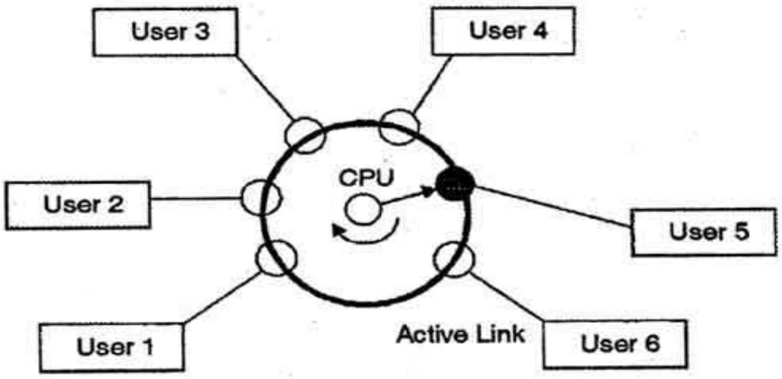
Fig: Multi-processing O S.

3. Time sharing OS.

A time sharing system allows many users to share the computer resources simultaneously. In other words, time sharing refers to the allocation of computer resources in time slots to several programs simultaneously. For example a mainframe computer that has many users logged on to it. Each user uses the resources of the mainframe -i.e. memory, CPU etc. The users feel that they are exclusive user of the CPU, even though this is not possible with one CPU i.e. shared among different users.

The time sharing systems were developed to provide an interactive use of the computer system. A time shared system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer. It allows many users to share the computer resources simultaneously. As the system switches rapidly from one user to the other, a short time slot is given to each user for their executions.

The time sharing system provides the direct access to a large number of users where CPU time is divided among all the users on scheduled basis. The OS allocates a set of time to each user. When this time is expired, it passes control to the next user on the system. The time allowed is extremely small and the users are given the impression that they each have their own CPU and they are the sole owner of the CPU. This short period of time during that a user gets attention of the CPU; is known as a time slice or a quantum. The concept of time sharing system is shown in figure.



In above figure the user 5 is active but user 1, user 2, user 3, and user 4 are in waiting state whereas user 6 is in ready status.

As soon as the time slice of user 5 is completed, the control moves on to the next ready user i.e. user 6. In this state user 2, user 3, user 4, and user 5 are in waiting state and user 1 is in ready state. The process continues in the same way and so on.

The time-shared systems are more complex than the multi-programming systems. In time-shared systems multiple processes are managed simultaneously which requires an adequate management of main memory so that the processes can be swapped in or swapped out within a short time.

Note: The term 'Time Sharing' is no longer commonly used; it has been replaced by 'Multitasking System'.

4. Network OS.
   Refers to software that implements an operating system of some kind that is oriented to computer networking. For example, one that runs on a server and enables the server to manage data, users, groups, security, applications, and other networking functions. The network operating system is designed to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.

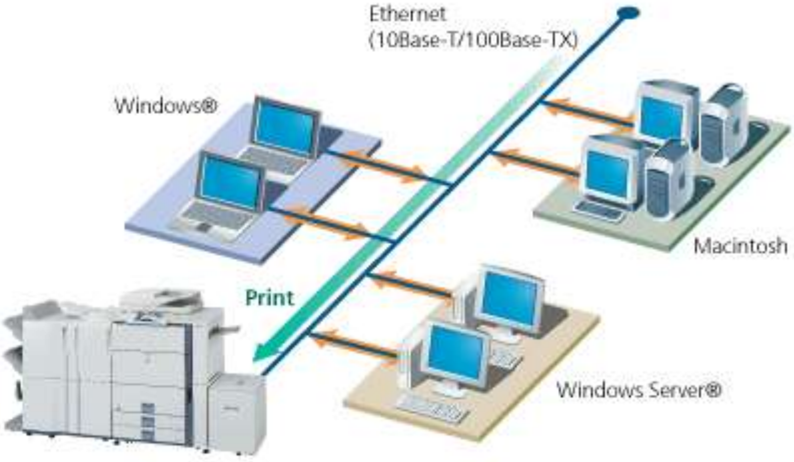
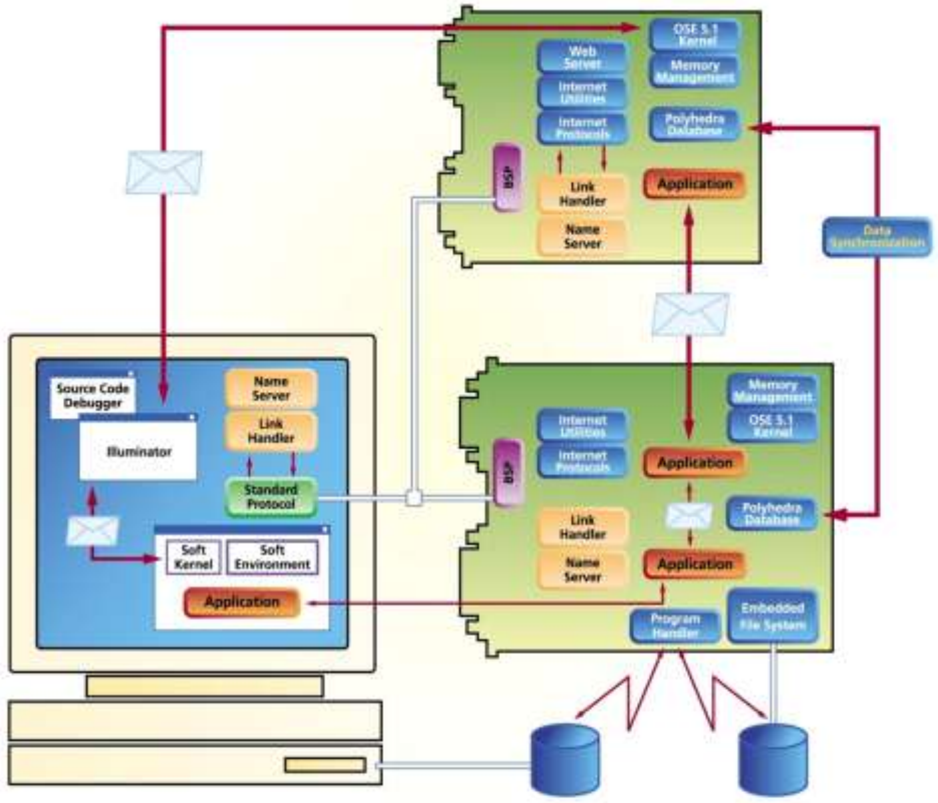


Fig: Network O S.

5. Real time OS.

(RTOS) is real time operating system intended to serve real-time application requests. It must be able to process data as it comes in, typically without buffering delays. Processing time requirements (including any OS delay) are measured in tenths of seconds or shorter.



OSE Real-Time Operating System v. 5.1

Example: A common example of an RTOS is an HDTV receiver and display. It needs to read a digital signal, decode it and display it as the data comes in. Any delay would be noticeable as jerky or pixelated video and/or garbled audio.

Note: HDTV is high definition TV

2. Application software: -

Is all the computer software that causes a computer to perform useful tasks (compare with computer viruses) beyond the running of the computer itself? A specific instance of such software is called a **software application**, **program**, **application** or **app**.

-All other software that runoff computer, for example: -

- *Word processing*: (Word) application for producing letters, (MS – word).
- *Spread sheet application*: (Excel) for doing accounts of working with number, (MS – Excel).
- *Data base application*: (Access) for organizing large amounts of information, (MS – Access).
- *Presentation application*: (Power Point) for producing picture & manuals, (MS – Power Point).

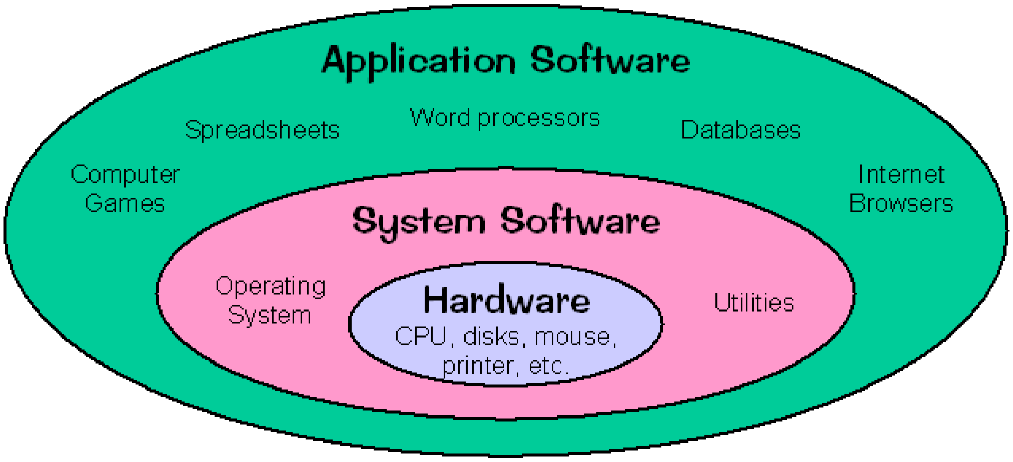


Fig: Application software

4.3: Types of software according to source: -

1. Commercial software.
2. Shareware software.
3. Freeware software.
4. Public domain software.

Computer software comes in **three** different flavors: ***freeware, shareware, and commercial software***.

Freeware**:** is free to use and does not require any payment from the user. ***Shareware:*** is also free to use, but typically limits the program's features or the amount of time the software can be used unless the user purchases the software.

Commercial software: requires payment before it can be used, but includes all the program's features, with no restrictions or time limits.

Commercial software programs typically come in a physical box, which is what you see displayed in retail stores. While it's true that the software boxes are not as big as they used to be, they still contain the software CD or DVD and usually a "getting started" manual along with a registration key used for registering the product. Most commercial software programs ask that the user register the program so the company can keep track of its authorized users. Some commercial software programs, such as newer versions of Microsoft and Adobe programs, require the user to register the programs in order to continue using them after 30 days.

While most commercial software programs are sold in the physical box, many software titles are now available as downloads. These downloads are typically made available from the company's website. The user pays for the program directly on the website and instead of receiving the software in the mail; the user downloads it to his computer. Another popular way of purchasing commercial software online is simply paying for a registration key, which unlocks the features of a shareware program. This upgrades the shareware program to the commercial version, which removes any feature limitations from the shareware version.

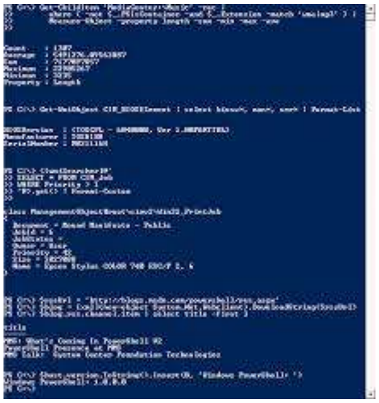
Public domain software:

Software refers to software that is not limited by having a copyright. It can thus be freely used, copied, or altered, because no one owns the rights to restrict its use. There are plenty of public domain programs, but these are often confused with several other types of software, called freeware, which can be obtained for free, or shareware, which is usually obtained for a very small fee. Public domain software is the only type in most cases to which no limitations apply. Freeware may be copyrighted and shareware certainly is. This means it is not essentially in the public domain. Someone else can’t reproduce the software, copy it for other people, or copy it to sell without violating copyright laws.

4.4: Interfaces: -

Is the way that the user used to deal with the computer or communicate his commands to the computer. Or In computer science, an interface is the point of interaction with software, or computer hardware, or with peripheral devices such as a computer monitor or a keyboard.

1. Command line interface. (CLI). is a means of interacting with a computer program where the user (or client) issues commands to the program in the form of successive lines of text (command lines).



```
PS C:\> Get-ChildItem 'MediaCenter:\Music' -rec |
>>       where { -not $_.PSIsContainer -and $_.Extension -match 'wma|mp3' } |
>>       Measure-Object -property length -sum -min -max -ave
>>

Count    : 1387
Average  : [illegible]
Sum      : [illegible]
Maximum  : [illegible]
Minimum  : [illegible]
Property : Length

PS C:\> Get-WmiObject CIM_BIOSElement | select biosv*, man*, ser* | Format-List

BIOSVersion  : [illegible]
Manufacturer : [illegible]
SerialNumber : [illegible]

PS C:\> [WmiSearcher]@'
>> SELECT * FROM CIM_Job
>> WHERE Priority > 1
>> '@.get() | Format-Custom
>>

class ManagementObject#root\cimv2\Win32_PrintJob
{
  Document = [illegible]
  JobId = [illegible]
  JobStatus =
  Owner = [illegible]
  Priority = 42
  Size = [illegible]
  Name = Epson Stylus COLOR 740 ESC/P 2, 6
}

PS C:\> $rssUrl = 'http://blogs.msdn.com/powershell/rss.aspx'
PS C:\> $blog = [xml](new-object System.Net.WebClient).DownloadString($rssUrl)
PS C:\> $blog.rss.channel.item | select title -first 3

title
-----
MMS: What's Coming In PowerShell V2
PowerShell Presence at MMS
MMS Talk: System Center Foundation Technologies

PS C:\> $host.version.ToString().Insert(0, 'Windows PowerShell: ')
Windows PowerShell: 1.0.0.0
PS C:\>
```

Fig: command line interface

2. Graphic user interface (GUI). is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces,

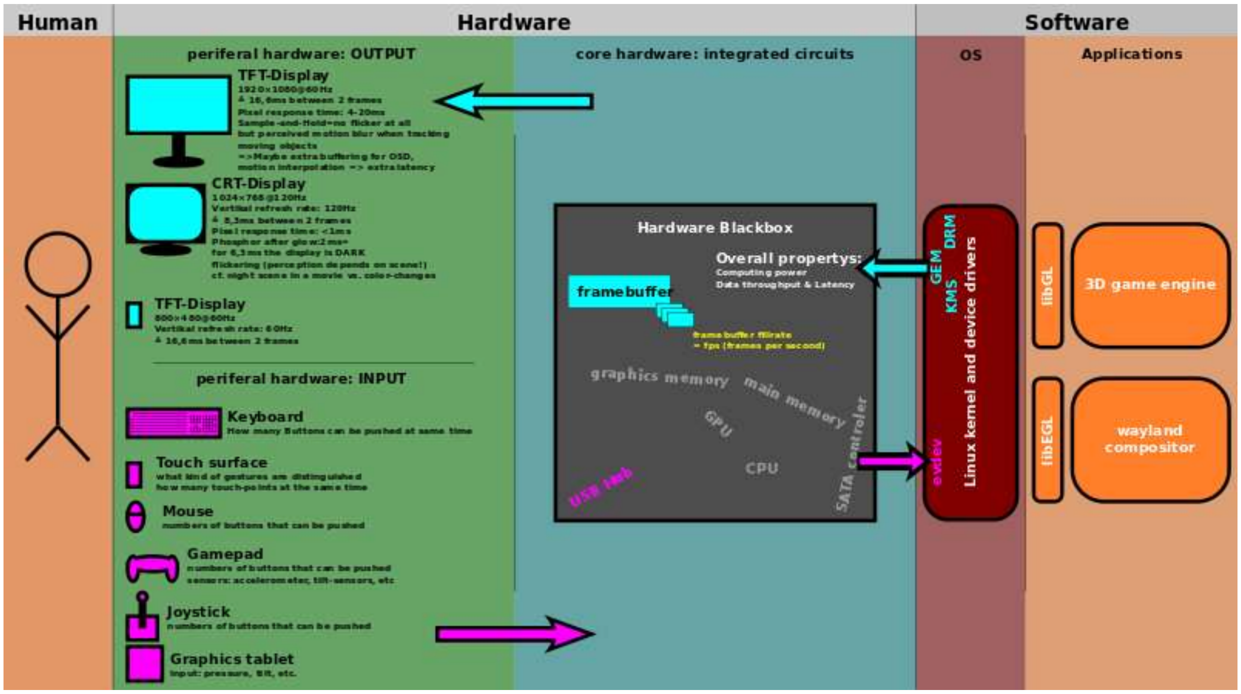

Human
Hardware
Software
periferal hardware: OUTPUT
core hardware: integrated circuits
OS
Applications
TFT-Display
1920×1080@60Hz
≙ 16,6ms between 2 frames
Pixel response time: 4-20ms
Sample-and-Hold=no flicker at all
but perceived motion blur when tracking
moving objects
=>Maybe extra buffering for OSD,
motion interpolation => extra latency
CRT-Display
1024×768@120Hz
Vertikal refresh rate: 120Hz
≙ 8,3ms between 2 frames
Pixel response time: <1ms
Phosphor after glow:2ms=
for 6,3ms the display is DARK
flickering (perception depends on scene!)
cf. night scene in a movie vs. color-changes
TFT-Display
800×480@60Hz
Vertikal refresh rate: 60Hz
≙ 16,6ms between 2 frames
periferal hardware: INPUT
Keyboard
How many Buttons can be pushed at same time
Touch surface
what kind of gestures are distinguished
how many touch-points at the same time
Mouse
numbers of buttons that can be pushed
Gamepad
numbers of buttons that can be pushed
sensors: accelerometer, tilt-sensors, etc
Joystick
numbers of buttons that can be pushed
Graphics tablet
input: pressure, tilt, etc.
Hardware Blackbox
Overall propertys:
Computing power
Data throughput & Latency
framebuffer
frame buffer fillrate
= fps (frames per second)
graphics memory
main memory
GPU
CPU
SATA controler
USB hub
DRM
GEM
KMS
Linux kernel and device drivers
evdev
libGL
libEGL
3D game engine
wayland compositor

Fig: The graphical user interface is presented /displayed on the screen. It is the result of processed user input and usually the primary interface for human-machine interaction. The Touch user interfaces popular on small mobile devices are an overlay of the visual output to the visual input.

4.5: System development: -

The steps that develop any system may be computer steps of system development: -

1. Define requirement.
2. Assess feasibility.
3. System analysis.
4. Product development.
5. Training the client.
6. Testing the system.
7. Monitoring & maintain the system.

4.6: Multimedia: -

Is the use of different media in computer programs: text, graphics, audio, video & animation, it is used in presentations, education, learning, games & commercial purposes.

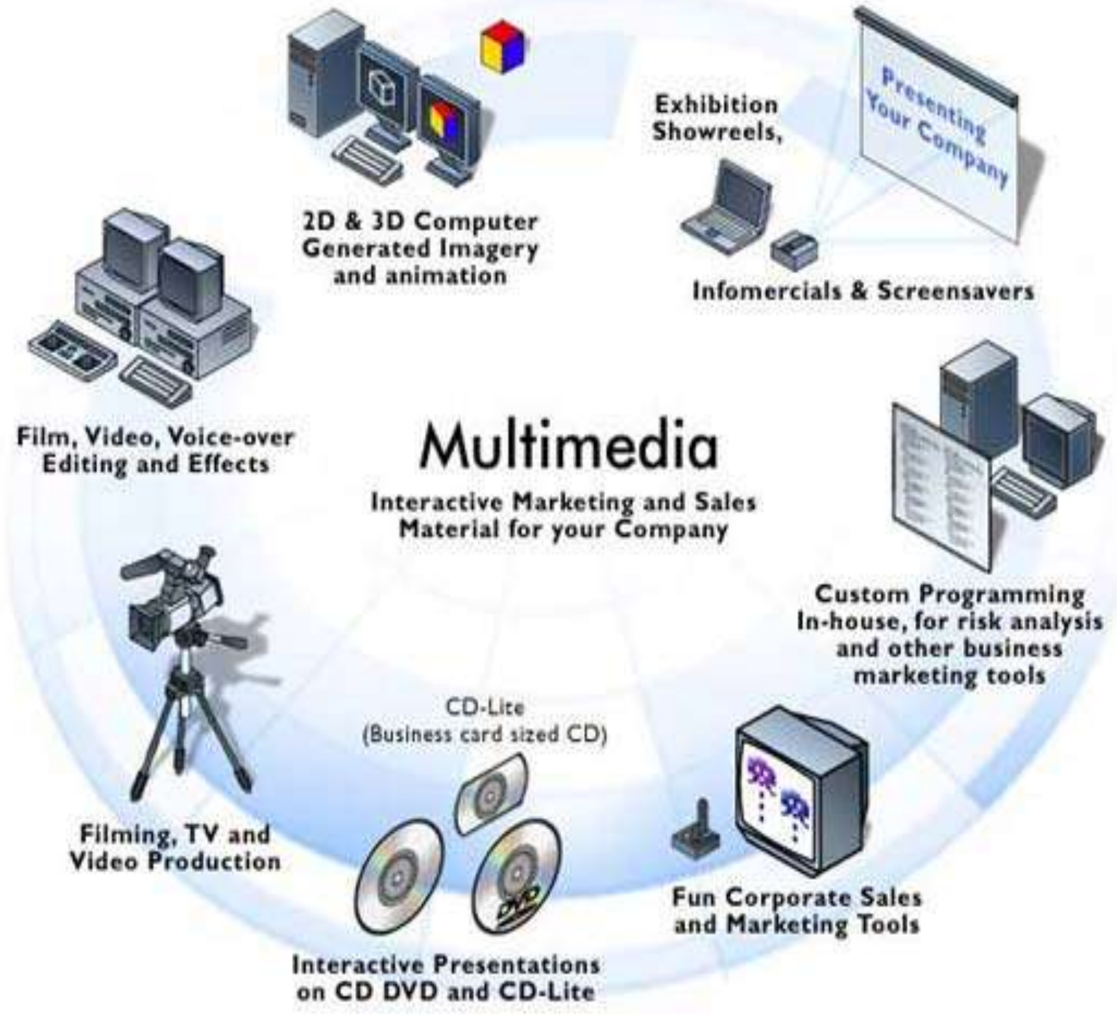

2D & 3D Computer Generated Imagery and animation
Exhibition Showreels,
Presenting Your Company
Infomercials & Screensavers
Film, Video, Voice-over Editing and Effects
Multimedia
Interactive Marketing and Sales Material for your Company
Custom Programming In-house, for risk analysis and other business marketing tools
Filming, TV and Video Production
CD-Lite (Business card sized CD)
DVD
Fun Corporate Sales and Marketing Tools
Interactive Presentations on CD DVD and CD-Lite

Fig: different kind of multimedia

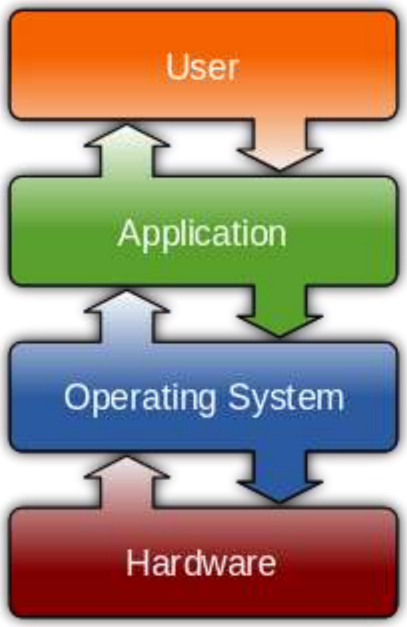


Fig show: A layer structure showing where the operating system software and application software are situated while running on a typical desktop computer.

H.W.: -

1. What is an operating system? Give examples.
2. What are application programs?
3. Explain the system development cycle?