Unit 1

Introduction to Operating system

Operating System: -

” It is an interface between computer user and computer hardware it also provides basis for application program.”

“A program that acts as an intermediary between a user of a computer and the computer hardware.”

Goals of the Operating system: -

1) Execute the user programs.

2) Help to solve problem.

3)Make the system convenient to use.

4) Work efficiently.

Because an operating system is large and complex, it must be created piece by piece. Each of these pieces should be a well-delineated portion of the system, with carefully defined inputs, outputs, and functions. In this chapter, we provide a general overview of the major components of a contemporary computer system as well as the functions provided by the operating system.

Personal computer (PC) provides environment for business, complex gaming.

Mobile computer (Smartphone) provides environment to execute programs.

Operating system components can be divided as: -

1)Hardware

2)Operating system

3)Application programmer

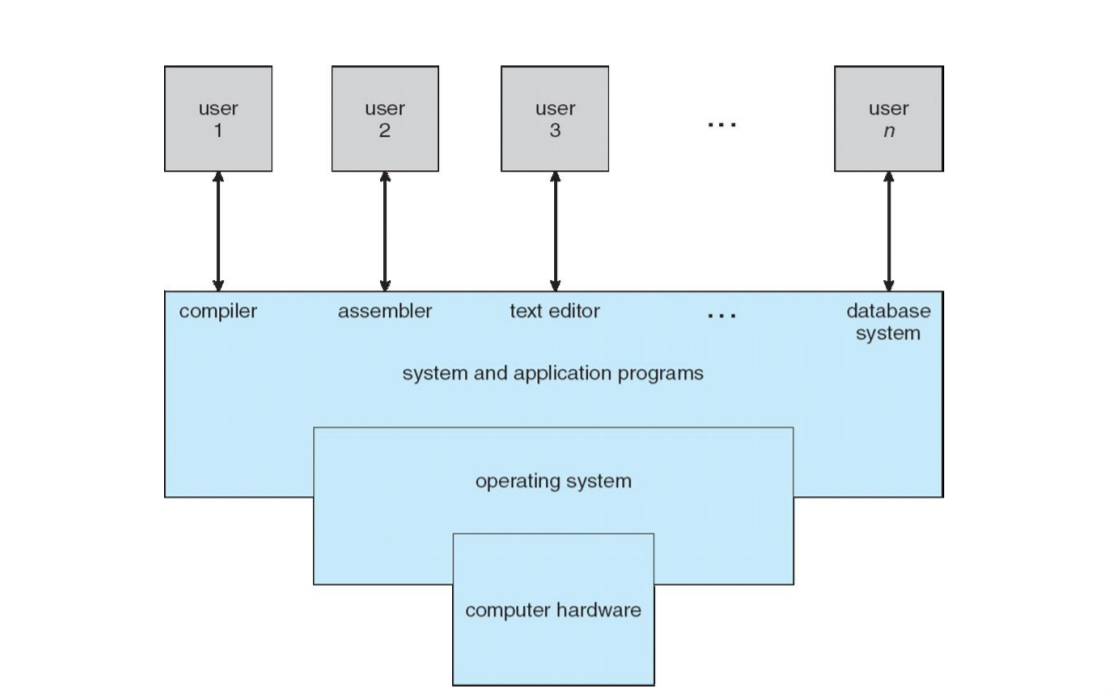
4)User

Hardware: - basic computing resources.

Operating system: - conveys instruction between hardware and application program.

Application Program: - handles resources and deals with the problem solving.

User: - person or machine that provides instruction.

Operating system components

The operating system provides the means for proper use of these resources in the operation of the computer system. An operating system is similar to a government. Like a government, it performs no useful function by itself. It simply provides an environment within which other programs can do useful work.

To understand the system better let's go through the following viewpoints: -

1) User view

2) System view

User view: -

Case 1: -

Most computer users sit in front of a PC, consisting of a monitor, keyboard, mouse, and system unit. Such a system is designed for one user to monopolize its resources. The goal is to maximize the work (or play) that the user is performing.

For such user the operating system is designed with the concept of ease to use without recognition of the resource's utilization. (how the hardware or software is shared)

Case 2: -

A user sits at a terminal connected to a mainframe or a minicomputer. Other users are accessing the same computer through other terminals. These users share resources and may exchange information.

For such user the operating system is designed to maximize resource utilization—to assure that all available CPU time, memory, and I/O are used efficiently and that no individual user takes more than her fair share.

Case 3: -

A user sits at workstations connected to networks of

other workstations and servers.

For such user operating system is designed to compromise between individual usability and resource utilization.

Case 4: -

Mobile computers are standalone units for individual users. Quite often, they are connected to networks through cellular or other wireless technologies. Increasingly, these mobile devices are replacing desktop and laptop computers for people who are primarily interested in using computers for e-mail and web browsing.

For user interface for mobile computers generally features a touch screen, where the user interacts with the system by pressing and swiping fingers across the screen rather than using a physical keyboard and mouse.

System View: -

A computer system has many resources that may be required to solve a problem: CPU time, memory space, file-storage space, I/O devices, and so on.

The operating system acts as the manager of these resources. Facing numerous and possibly conflicting requests for resources, the operating system must decide how to allocate them to specific programs and users so that it can operate the computer system efficiently and fairly.

As we have seen, resource allocation is especially important where manyusers access the same mainframe or minicomputer.

A slightly different view of an operating system emphasizes the need to control the various I/O devices and user programs. An operating system is a control program.

A control program manages the execution of user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of I/O devices.

In the 1960s, Moore’s Law predicted

that the number of transistors on an integrated circuit would double every eighteen months, and that prediction has held true. Computers gained in functionality and shrunk in size, leading to a vast number of uses and a vast number and variety of operating systems.

The fundamental goal of computer systems is to execute user programs and to make solving user problems easier. Computer hardware is constructed toward this goal. Since bare hardware alone is not particularly easy to use, application programs are developed.

The common functions of controlling and allocating resources are then brought together into one piece of software: the operating system. the operating system is the one program running at all times on the computer—usually called the kernel. (Along with the kernel, there are two other types of programs: system programs, which are associated with the operating system but are not necessarily part of the kernel, and application programs, which include all programs not associated with the operation of the system.)

Mobile operating systems often include not only a core kernel but also middleware (a set of software frameworks that provide additional services to application developers). For example, each of the two most prominent mobile operating systems—Apple’s iOS and Google’s Android—features a core kernel along with middleware that supports databases, multimedia, and graphics.

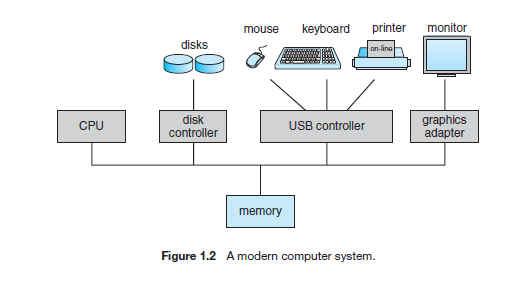
Computer-System Operation: -

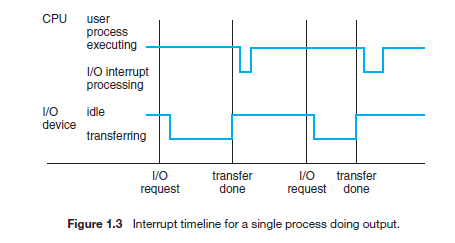
For a computer to start running—for instance, when it is powered up or rebooted—it needs to have an initial program to run. This initial program, or bootstrap program.

It is stored within the computer hardware in read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM), known by the general term firmware.

Once the kernel is loaded and executing, it can start providing services to the system and its users. Some services are provided outside of the kernel, by system programs that are loaded into memory at boot time to become system processes, or system daemons that run the entire time the kernel is running.

The occurrence of an event is usually signaled by an interrupt from either the hardware or the software. Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by way of the system bus. Software may trigger an interrupt by executing a special operation called a system call (also called a monitor call).





Storage Structure: -

General-purpose computers run most of their programs from rewritable memory, called main memory or RAM. Main memory commonly is implemented in a semiconductor technology called dynamic random-access memory (DRAM).

We want the programs and data to reside in main memory permanently. This arrangement usually is not possible for the following two reasons:

1. Main memory is usually too small to store all needed programs and data permanently.

2. Main memory is a volatile storage device that loses its contents when power is turned off or otherwise lost.

Volatile storage loses its contents when the power to the device is removed. In the absence of expensive battery and generator backup systems, data must be written to nonvolatile storage for safekeeping.

Storage systems organized in hierarchy

Speed

Cost

Volatility

Caching – copying information into faster storage system; main memory can be viewed as a cache for secondary storage

Device Driver for each device controller to manage I/O Provides uniform interface between controller and kernel.

Caching: -(I/O structure)

Important principle, performed at many levels in a computer (in hardware, operating system, software)

Information in use copied from slower to faster storage temporarily

Faster storage (cache) checked first to determine

if information is there If it is, information used directly from the cache (fast)

If not, data copied to cache and used there

Cache smaller than storage being cached

Cache management important design problem

Cache size and replacement policy.

Used for high-speed I/O devices able to transmit information at close to memory speeds

Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention

Only one interrupt is generated per block, rather than the one interrupt per byte.

Computer system Architecture: -

There are 3 types of processing system

1) Single processor system

2) Multiprocessor system

3)Clustered system

Single processor system: -

This type of processor has one main CPU and can execute instruction or program for the user. These processors have special process processor to perform I/O instruction rapidly.

Multiprocessor system: - multiprocessor systems also known as parallel systems or multicore systems. These type of system have more than two processor to get the work done faster.

1)Increased throughput this means increase the number of processors and get the work done faster.

2)Economy of scale Instead of buying number of singles processing system buy the processor and get the function design done so you can share all the peripheral devises.

3)Increased reliability distribute the function properly so that even if one of the system crashes others can work.

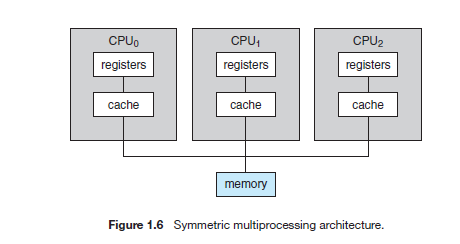
The ability to continue providing service proportional to the level of surviving hardware is called graceful degradation. Some systems go beyond graceful degradation and are called fault tolerant.

Example. The HP Nonstop (formerly Tandem) system uses both hardware and software duplication to ensure continued operation despite faults. The system consists of multiple pairs of CPUs, working in lockstep.

The multiple-processor systems are of two types:

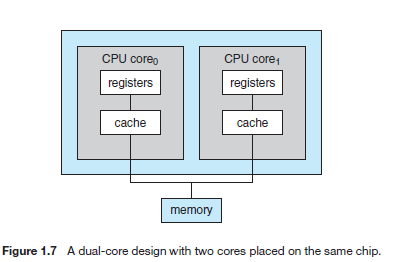
Asymmetric multiprocessing. (Boss-Worker)

Symmetric multiprocessing (SMP). (Each processor is Boss)



Multiprocessing can cause a system to change its memory access model from uniform memory access (UMA) to non-uniform memory access (NUMA). UMA is defined as the situation in which access to any RAM from any CPU takes the same amount of time. With NUMA, some parts of memory may take longer to access than other parts, creating a performance penalty.

CPU design is to include multiple computing cores on a single chip. Such multiprocessor systems are termed multicore. But not all multiprocessor system has multicore.



blade servers are a relatively recent development in which multiple processor boards, I/O boards, and networking boards are placed in the same chassis. The difference between these and traditional multiprocessor systems is that each blade-processor board boots independently and runs its own operating system.

Clustered Systems: - Clustered systems differ from the multiprocessor systems they are composed of two or more individual systems—or nodes—joined together. Such systems are considered loosely coupled.

Clustered computers share storage and are closely linked via a local-area network (LAN).

Clustering can be structured asymmetrically or symmetrically.

* In asymmetric clustering, one machine is in hot-standby mode while the other is running the applications. The hot-standby host machine does nothing but monitor the active server. If that server fails, the hot-standby host becomes the active server.
* In symmetric clustering, two or more hosts are running applications and are monitoring each other. This structure is obviously more efficient, as it uses all of the available hardware. However it does require that more than one application be available to run.

Parallelization divides a program into separate components that run in parallel on individual computers in the cluster.

Other form of clusters: -

Parallel clusters allow multiple hosts to access the same data on shared storage.

