PART – A

1. **What is OS?**

An operating system is software, consisting of programs and data that runs on computers and manages computer hardware resources and provides common services for efficient execution of various application software.

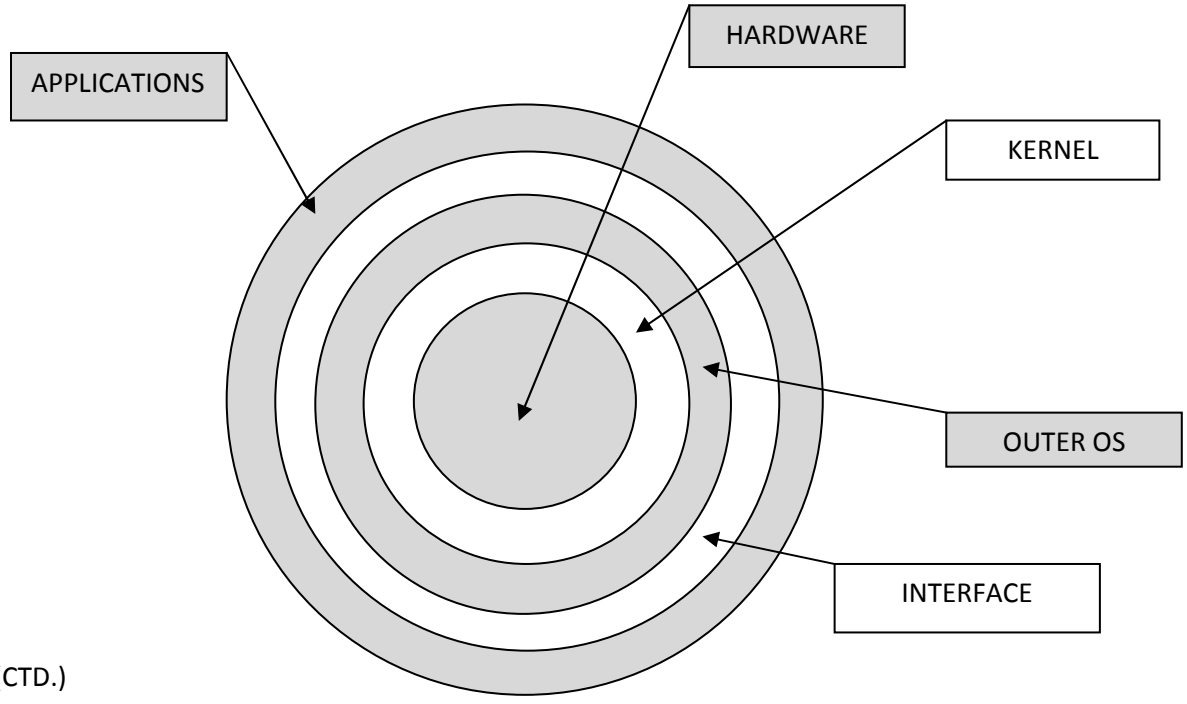
Operating systems are found on almost any device that contains a computer—from cellular phones and video game consoles to supercomputers and web servers.

2. **What is single processor?**

It is a processor that has only one core, so it can only start one operation at a time. It can however in some situations start a new operation before the previous one is complete. Originally all processors were single core.

To improve efficiency, processors commonly utilize pipelines internally, which allow several instructions to be processed together; however, they are still consumed into the pipeline one at a time. Multi core processors were introduced later, when increasing the clock speeds further was too hard. These new multi core processors are two processing units combined into one.

3. **What are the methods of operating system structure?**



(CTD.)

1. **HARDWARE**: All those I/O peripherals and all physical elements of a system.
2. **KERNEL**: The bottom-most layer of software present on a machine and the only one with direct access to the hardware. *The code in the kernel is the most 'trusted' in the system - and all requests to do anything significant must go via the kernel. It provides the most key facilities and functions of the system.*
3. **OUTER OS:** These perform less critical functions - *for example*, the graphics system which is ultimately responsible for what you see on the screen.
4. **INTERFACE:** The interface provides a mechanism for you to interact with the computer.
5. **APPLICATIONS:** These are the things which do the actual work. (For example Office or simple (for example the "ls" command commonly found on unix and Linux systems that lists files in a directory or folder).

4. Define processor.

The processor unit (CPU) is the portion of a computer system that carries out the instructions of a computer program, and is the primary element carrying out the computer's functions. The central processing unit carries out each instruction of the program in sequence, to perform the basic arithmetical, logical, and input/output operations of the system. This term has been in use in the computer industry at least since the early 1960s. The form, design and implementation of CPUs have changed dramatically since the earliest examples, but their fundamental operation remains much the same.

5. What is meant by system calls?

A **system call** is how a program requests a service from an operating system's kernel that it does not normally have permission to run. System calls provide the interface between a process and the operating system. Most operations interacting with the system require permissions not available to a user level process, e.g. I/o performed with a device present on the system, or any form of communication with other processes requires the use of system calls.

_____

_____

PART – B

1. Explain about virtual machines.

A virtual machine (VM) is a software implementation of a machine (i.e. a computer) that executes programs like a physical machine. Virtual machines are separated into two major categories, based on their use and degree of correspondence to any real machine. A **system virtual machine** provides a complete system platform which supports the execution of a complete operating system (OS). In contrast, a **process virtual machine** is designed to run a single program, which means that it supports a single process. An essential characteristic of a virtual machine is that the software running inside is limited to the resources and abstractions provided by the virtual machine—it cannot break out of its virtual world.

System virtual machines:

System virtual machines (sometimes called **hardware virtual machines**) allow the sharing of the underlying physical machine resources between different virtual machines, each running its own operating system.

The main advantages of VMs are:

- Multiple OS environments can co-exist on the same computer, in strong isolation from each other
- The virtual machine can provide an instruction set architecture (ISA) that is somewhat different from that of the real machine
- Application provisioning, maintenance, high availability and disaster recovery

The main disadvantages of VMs are:

- A virtual machine is less efficient than a real machine when it accesses the hardware indirectly
- When multiple VMs are concurrently running on the same physical host, each VM may exhibit a varying and unstable performance (speed of execution, and not results), which highly depends on the workload imposed on the system by other VMs, unless proper techniques are used for temporal isolation among virtual machines.

The desire to run multiple operating systems was the original motivation for virtual machines, as it allowed time-sharing a single computer between several single-tasking OS. In some respects, a system virtual machine can be considered a generalization of the concept of virtual memory that historically preceded it. IBM's CP/CMS, the first systems to allow full virtualization, implemented time sharing by providing each user with a single-user operating system, the CMS. Unlike virtual memory, a system virtual machine allowed the user to use privileged instructions in their code. This approach had certain advantages, for instance it allowed users to add input/output devices not allowed by the standard system.

Process virtual Machine:

A process VM, sometimes called an application virtual machine, runs as a normal application inside an OS and supports a single process. It is created when that process is started and destroyed when it exits. Its purpose is to provide a platform-independent programming environment that abstracts away details of the underlying hardware or operating system, and allows a program to execute in the same way on any platform.

A process VM provides a high-level abstraction — that of a high-level programming language (compared to the low-level ISA abstraction of the system VM). Process VMs are implemented using an interpreter; performance comparable to compiled programming languages is achieved by the use of just-in-time compilation.

This type of VM has become popular with the Java programming language, which is implemented using the Java virtual machine. Other examples include the Parrot virtual machine, which serves as an abstraction layer for several interpreted languages, and the .NET Framework, which runs on a VM called the Common Language Runtime.

A special case of process VMs are systems that abstract over the communication mechanisms of a (potentially heterogeneous) computer cluster. Such a VM does not consist of a single process, but one process per physical machine in the cluster. They are designed to ease the task of programming parallel applications by letting the programmer focus on algorithms rather than the communication mechanisms provided by the interconnect and the OS. They do not hide the fact that communication takes place, and as such do not attempt to present the cluster as a single parallel machine.

A virtual machine can also be a virtual environment, which is also known as a virtual private server. A virtual environment is used for running programs at the user level. Therefore, it is used solely for applications and not for drivers or operating system kernels.

2. Define cluster system and multi processor.

Cluster system

A **computer cluster** is a group of linked computers, working together closely thus in many respects forming a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks. Clusters are usually deployed to improve performance and availability over that of a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.

Categories of clusters

- **High-availability (HA) clusters:** High-availability clusters (also known as Failover Clusters) are implemented primarily for the purpose of improving the availability of services that the cluster provides. They operate by having redundant nodes, which are then used to provide service when system components fail. The most common size for an HA cluster is two nodes, which is the minimum requirement to provide redundancy. HA cluster implementations attempt to use redundancy of cluster components to eliminate single points of failure.
- **Load-balancing clusters:** Load-balancing is when multiple computers are linked together to share computational workload or function as a single virtual computer. Logically, from the user side, they are multiple machines, but function as a single virtual machine. Requests initiated from the user are managed by, and distributed among, all the standalone computers to form a cluster. This results in balanced computational work among different machines, improving the performance of the cluster systems.
- **Compute clusters:** Often clusters are used primarily for computational purposes, rather than handling IO-oriented operations such as web service or databases. For instance, a cluster might support computational of weather or vehicle crashes. The primary distinction within computer clusters is how tightly-coupled the individual nodes are. For instance, a single computer job may require frequent communication among nodes - this implies that the cluster shares a dedicated network, is densely located, and probably has homogenous nodes. This cluster design is usually referred to as Beowulf Cluster. The other extreme is where a computer job uses one or few nodes, and needs little or no inter-node communication. This latter category is sometimes called "Grid" computing.

Notable implementations

Consumer game consoles

Due to the increasing computing power of each generation of game consoles, a novel use has emerged where they are repurposed into High-performance computing (HPC) clusters. Some examples of game console clusters are Sony PlayStation clusters and Microsoft Xbox clusters. It has been suggested on a news website that countries which are restricted from buying supercomputing technologies may be obtaining game systems to build computer clusters for military use.

Virtual Machines

A **processor** is the unit that reads and executes program instructions, which are fixed-length (typically 32 or 64 bit) or variable-length chunks of data. The data in the instruction tells the processor what to do. The instructions are very basic things like reading data from memory or sending data to the user display, but they are processed so rapidly that we experience the results as the smooth operation of a program.

A **multi-core processor** is composed of two or more independent cores. One can describe it as an integrated circuit which has two or more individual processors (called *cores* in this sense). Manufacturers typically integrate the cores onto a single integrated circuit die (known as a chip multiprocessor or CMP), or onto multiple dies in a single chip package. A **many-core** processor is one in which the number of cores is large enough that traditional multi-processor techniques are no longer efficient— largely due to issues with congestion supplying sufficient instructions and data to the many processors. This threshold is roughly in the range of several tens of cores and probably requires a network on chip. The amount of performance gained by the use of a multi-core processor depends very much on the software algorithms and implementation.

Advantages

The proximity of multiple CPU cores on the same die allows the cache coherency circuitry to operate at a much higher clock-rate than is possible if the signals have to travel off-chip. Combining equivalent CPUs on a single die significantly improves the performance of cache snoop (alternative: Bus snooping) operations. Put simply, this means that signals between different CPUs travel shorter distances, and therefore those signals degrade less. These higher-quality signals allow more data to be sent in a given time period, since individual signals can be shorter and do not need to be repeated as often.

The largest boost in performance will likely be noticed in improved response-time while running CPU-intensive processes, like antivirus scans, ripping/burning media (requiring file conversion), or file searching. For example, if the automatic virus-scan runs while a movie is being watched, the application running the movie is far less likely to be starved of processor power, as the antivirus program will be assigned to a different processor core than the one running the movie playback.

Assuming that the die can fit into the package, physically, the multi-core CPU designs require much less printed circuit board (PCB) space than do multi-chip SMP designs. Also, a dual-core processor uses slightly less power than two coupled single-core processors, principally because of the decreased power required to drive signals external to the chip. Furthermore, the cores share some circuitry, like the L2 cache and the interface to the front side bus (FSB). In terms of competing technologies for the available silicon die area, multi-core design can make use of proven CPU core library designs and produce a product with lower risk of design error than devising a new wider core-design. Also, adding more cache suffers from diminishing returns.

Disadvantages

Maximizing the utilization of the computing resources provided by multi-core processors requires adjustments both to the operating system (OS) support and to existing application software. Also, the ability of multi-core processors to increase application performance depends on the use of multiple threads within applications. The situation is improving: for example the Valve Corporation's Source engine offers multi-core support, and Crytek has developed similar technologies for CryEngine 2, which powers their game, *Crysis*. Emergent Game Technologies' Gamebryo engine includes their Floodgate technology which simplifies multicore development across game platforms. In addition, Apple Inc.'s latest OS, Mac OS X Snow Leopard has built-in multi-core facility called Grand Central Dispatch for Intel CPUs.

Integration of a multi-core chip drives chip production yields down and they are more difficult to manage thermally than lower-density single-chip designs. Intel has partially countered this first problem by creating its quad-core designs by combining two dual-core on a single die with a unified cache, hence any two working dual-core dies can be used, as opposed to producing four cores on a single die and requiring all four to work to produce a quad-core. From an architectural point of view, ultimately, single CPU designs may make better use of the silicon surface area than multiprocessing cores, so a development commitment to this architecture may carry the risk of obsolescence. Finally, raw processing power is not the only constraint on system performance. Two processing cores sharing the same system bus and memory bandwidth limits the real-world performance advantage. If a single core is close to being

memory-bandwidth limited, going to dual-core might only give 30% to 70% improvement. If memory bandwidth is not a problem, a 90% improvement can be expected. It would be possible for an application that used two CPUs to end up running faster on one dual-core if communication between the CPUs was the limiting factor, which would count as more than 100% improvement.
