DISTRIBUTED AND REAL OPERATING SYSTEM

Distributed System:

Multiprocessor system:
- General term for the use of two or more CPUs for a computer system.
- Can vary with context, mostly as a function of how CPUs are defined.
- The term multiprocessing is sometimes used to refer to the execution of multiple concurrent software processes in a system as opposed to a single process at any one instant.

Multiprogramming:
Multiprogramming is more appropriate to describe concept which is implemented mostly in softwares, whereas multiprocessing is more appropriate to describe the use of multiple hardware CPUs.
A system can be both multiprocessing and multiprogramming, only one of the two or neither of the two.

Processor Coupling:
Its the logical connection of the CPUs. Multiprocessor system have more than one processing unit sharing memory/peripherals devices. They have greater computing power and higher reliability. Multiprocessor system can be classified into two types:

1. Tightly coupled
2. Losely coupled(distributed). Each processor has its own memory and copy of the OS.

Tightly Coupled(Multiprocessor System): Tightly coupled multiprocessor system contain multiple CPUs that are connected at the bus level. Each processor is assigned a specific duty but processor work in close association, possibly sharing one memory module.
chip multiprocessors also known as multi-core computing involves more than one processor placed on a single chip and can be thought as the most extreme form of tightly coupled multiprogramming.
Dual core, Core-2 Duo, Intel Core I5 etc are the brand name used for various mid-range to high end consumers and business multiprocessor made by Intel.

Loosely Coupled(Distributed System):
Loosely coupled system often referred to as clusters are based on multiple stand-alone single or dual processors commodity computers interconnected via a high speed communication system. distributed system are connected via a distributed operating system.

Multiprocessor operating system:
Multiprocessor operating system aims to support high performance through the use of multiple CPUs. It consists of a set of processors that share a set of physical memory blocks over an interconnected

network. An important goal is to make the number of CPUs transparent to the application. Achieving such transparency is relatively easy because the communication between different (parts of) application uses the same primitives as those in uni-processor OS.
The idea is that all communication is done by manipulating data at the shared memory locations and that we only have to protect that data segment against simultaneous access. Protection is done through synchronization primitives like semaphores and monitors.

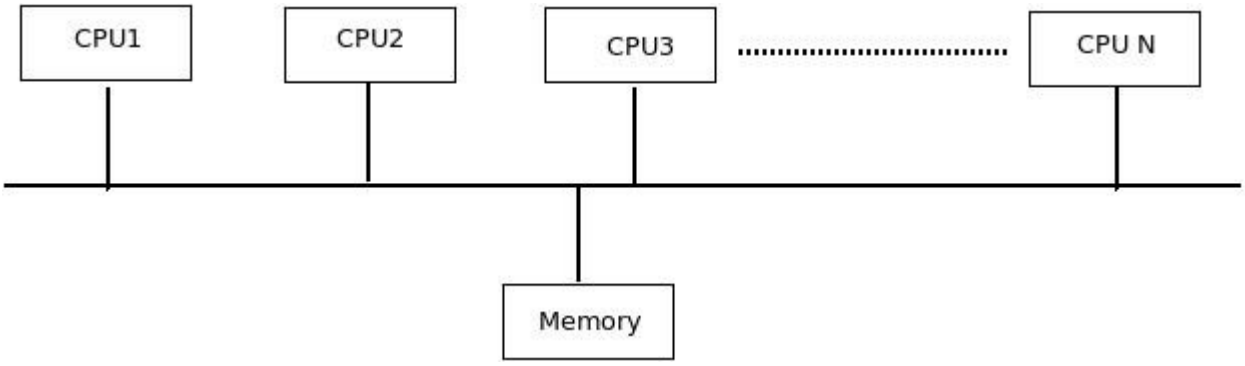

Fig: Multiprocessor System.

Distributed Operating System:

A recent trend in computer system is to distribute computation among several processors. In contrasts to the tightly coupled system the processors do not share memory or a clock. Instead, each processor has its own local memory. The processors communicate with one another through various communication lines such as computer network. Distributed operating system are the operating system for a distributed system(a network of autonomous computers connected by a communication network through a message passing mechanisms). A distributed operating system controls and manages the hardware and software resources of a distributed system. When a program is executed on a distributed system, user is not aware of where the program is executed or the location of the resources accessed.

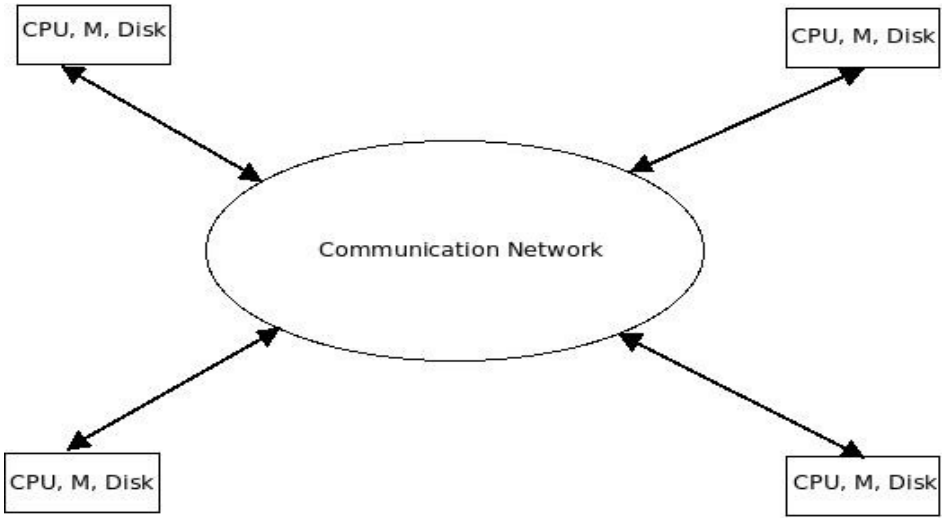

Fig: Architecture of a Distributed system.

Example of Distributed OS: Amoeba, Chorus, Alpha Kernel.

Real Time Operating System:

Primary objective of Real Time Operating System is to provide quick response time and thus to meet a scheduling deadline. User convenience and resource utilization are secondary concern to these systems. Real time systems has many events that must be accepted and processed in a short time or within certain deadline. Such applications include:
Rocket launching, flight control, robotics, real time simulation, telephone switching equipments etc.

Real time systems are classified into two categories:
a). Soft Real time System: If certain deadlines are missed then system continues its working with no failure but its performance degrade.

b). Hard Real time System: If any deadline is missed then system will fail to work or does not work properly. This system gurantees that critical task is completed on time.