AIM : To Study Distributed Operating System.

THEORY :

1. What is Distributed OS?

The Distributed Operating System is one that looks to its users like an ordinary centralized OS but runs on multiple, independent CPUs. The key concept is transparency here. The use of multiple processors should be invisible to the user. The user views the system as a virtual uniprocessor not as a collection of distinct machines.

A distributed OS provides the essential services and functionality required of an OS but adds attributes and particular configurations to allow it to support additional requirements such as increased scale and availability. To a user, a distributed OS works in a manner similar to a single-node, monolithic operating system. That is, although it consists of multiple nodes, it appears to users and applications as a single-node.

2. Features

- Connecting Users and Making Resources Available : The main goal of a distributed system is to make it easy for users to access remote resources, and to share them with other users in a controlled manner. Resources can be virtually anything, typical examples of resources are printers, storage facilities, data, files, web pages, and networks.
- Transparency : A distributed system that is capable of presenting itself to users and applications such that it is only a single computer system is called transparent.
- Scalability : A system is described as scalable if it remains effective when there is a significant increase in the number of resources and the number of users.
- Openness : The openness of DS is determined primarily by the degree to which new resource sharing services can be added and be made available for use by a variety of client programs. Services are specified through Interfaces, which are described in Interface Definition language. Open DS should be extensible.
- Reliability : The main goal of building distributed systems was to make them more reliable than single processor systems. The idea is that if some machine goes down, some other machine gets used to it.
- Performance : Building a transparent, flexible, reliable distributed system is useless if it is slow like molasses. In particular application on a distributed system, it should not deteriorate better than running some application on a single processor. Various performance metrics can be used. Response time is one, but so are throughput, system utilization, and amount of network capacity consumed.

3. Architecture

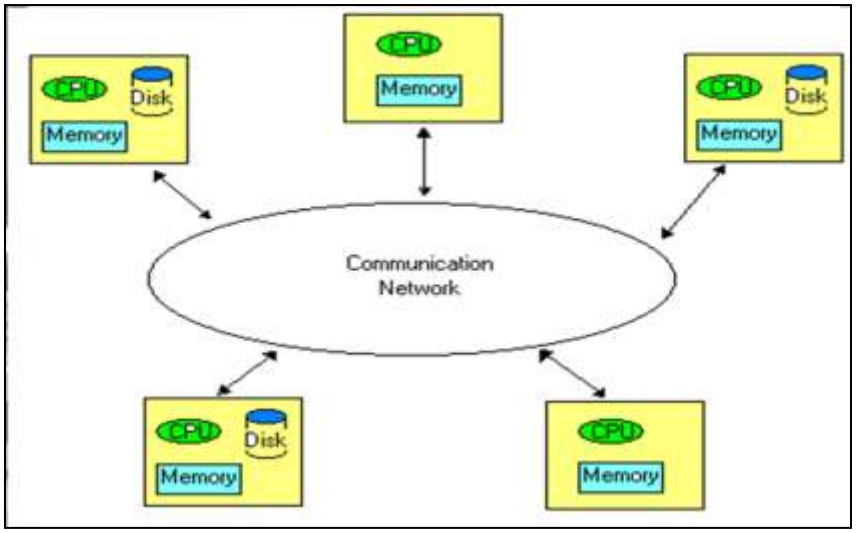

Figure 1 : Architecture of Distributed OS

4. Example of Distributed OS. Explain in Detail.

AMOEBA OS :

Amoeba is a distributed operating system developed by Andrew S. Tanenbaum and others at the Vrije Universiteit Amsterdam. The aim of the Amoeba project was to build a timesharing system that makes an entire network of computers appear to the user as a single machine. It is the fastest growing micro - kernel distributed system.

It is based on Processor Pool Model of Distributed computing where processors are pooled together to be shared by the users as needed and each processor has its own memory to load and run a system program.

Goals of Amoeba :

- Distribution : Connecting together many machines.
- Parallelism : Allowing individuals jobs to use multiple CPUs easily.
- Transparency : Having the collection of computer act like a single system.
- Performance : Achieving all of the above in an efficient manner.

Architecture :

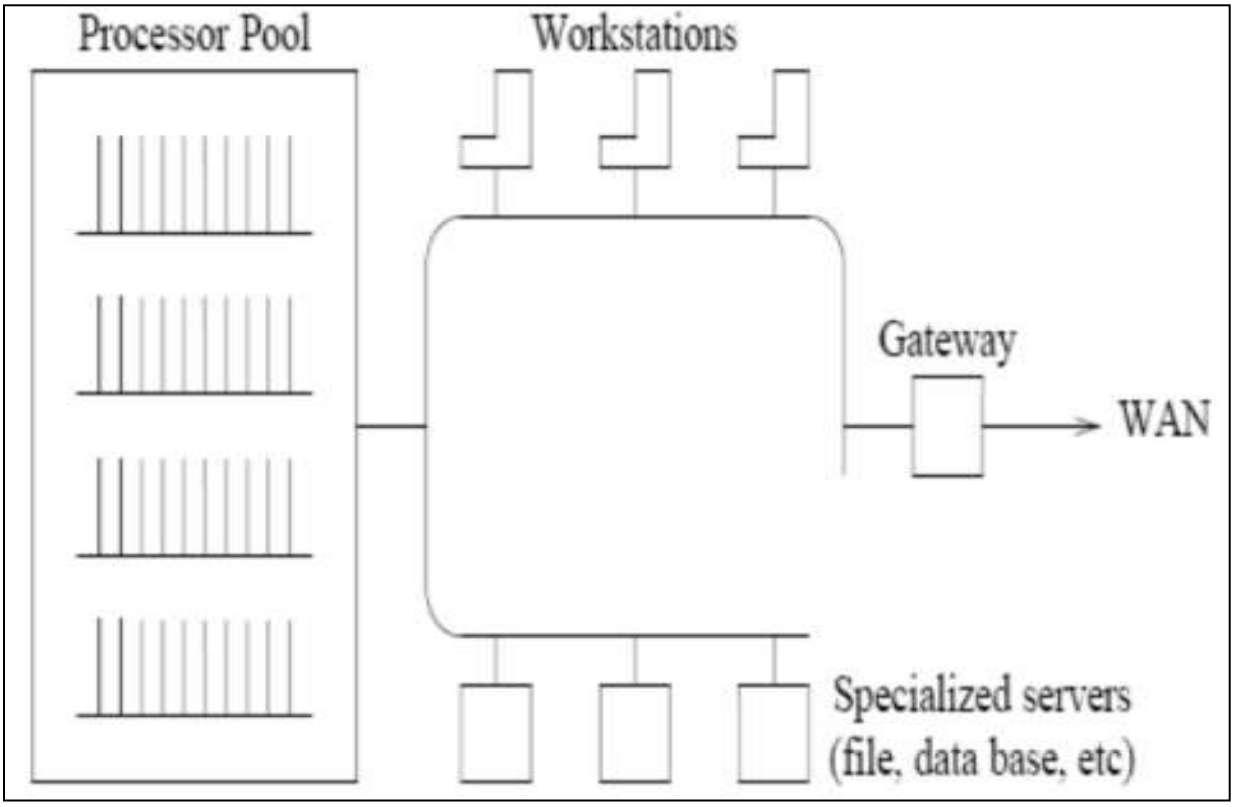

Figure 2 : Architecture of Amoeba OS

Advantages :

- Micro Kernel allows for other file systems to be created.
- Initially Unix emulation was added by developers to make it compatible with commonly used systems of that time so they have many common commands.
- System can be modified easily by a programmer according to requirements.
- Process is never swapped out.
- It efficiently makes the connected system work with maximum potential.

Disadvantages :

- Can only hold programs as large as its physical memory
- It has not had an official update in over 10 years.
- Designed to run on machines with large amounts of RAM and huge local disks.
- Process occupies contiguous segments in memory which may give rise to fragmentations.

Some Operating system Screenshots :

Figure 3 : Start Screen of Amoeba OS

Figure 4 : Log In Screen of Amoeba OS

5. Comparison of DOS with Network OS.

	Distributed OS	Network OS
Objective	Distributed OS manages the hardware resources.	Network OS provides local services to remote clients.
Communication	Communication is message-based or shared memory-based.	Communication is file-based, shared folder based.
Scalability	Distributed OS is less scalable. The process to add new hardware is complex.	Network OS is highly scalable. A new machine can be added very easily.
Fault tolerance	Distributed OS has very high fault tolerance.	Less fault tolerance as compared to distributed OS.
Autonomy	Distributed OS has a poor rate of autonomy.	Each machine can acts on its own thus autonomy is high.
Implementation	Distributed OS implementation is difficult.	Network OS-based systems are easy to built and maintain.
Operating System	Distributed OS-based nodes have the same copy of the operating system.	Network OS-based systems have their own copy of operating systems.

CONCLUSION :

From this experiment we understood the concept of Distributed operating systems. Distributed OS is system software over a collection of independent, networked, communicating, and physically separate computational nodes. We understood the features and architecture of Distributed OS. Later we created a Case study on Amoeba OS which is a distributed OS based on processor pool model. We saw its architecture, advantages, disadvantages and also how the OS looks. Finally we understood the difference between DOS and Network OS.