GOA COLLEGE OF ENGINEERING

"Bhausaheb Bandodkar Technical Education Complex"

Tutorial No: 2

Q1) What is the main difference between Distributed Operating System and Network Operating System?

The main difference between these two operating systems (Network Operating System and Distributed Operating System) is that in network operating system each node or system can have its own operating system on the other hand in distribute operating system each node or system have same operating system which is opposite to the network operating system.

A distributed operating system manages multiprocessors and homogeneous multicomputers. A network operating system connects different independent computers that each have their own operating system so that users can easily use the services available on each computer

DOS	NOS
Network Operating System's main objective is to provide the local services to remote client.	Distributed Operating System's main objective is to manage the hardware resources.
In Network Operating System, Communication takes place on the basis of files.	In Distributed Operating System, Communication takes place on the basis of messages and shared memory.
In Network Operating System, fault tolerance is less.	While in Distributed Operating System, fault tolerance is high

Q2) What are the primary tasks of a microkernel?

The microkernel is entirely responsible for the operating system's most significant services, which are as follows:

Inter-Process Communication

Interprocess communication refers to how processes interact with one another. A process has several threads. In the kernel space, threads of any process interact with one another. Messages are sent and received across threads using ports. At the kernel level, there are several ports like process port, exceptional port, bootstrap port, and registered port. All of these ports interact with user-space processes.

Memory Management

Memory management is the process of allocating space in main memory for processes. However, there

is also the creation of virtual memory for processes. Virtual memory means that if a process has a bigger size than the main memory, it is partitioned into portions and stored. After that, one by one, every part of the process is stored in the main memory until the CPU executes it.

CPU Scheduling

CPU scheduling refers to which process the CPU will execute next. All processes are queued and executed one at a time. Every process has a level of priority, and the process with the highest priority is performed out first. CPU scheduling aids in optimizing CPU utilization. In addition, resources are being used more efficiently. It also minimizes the waiting time.

Q3) Write 2 advantages of microkernel over monolithic kernel

- The microkernels are more secure than the monolithic kernels because the operating system is unchanged if a service fails in a microkernel. On the other hand, if a service fails in a monolithic kernel, the entire system fails.
- The microkernel is simple to extend as new services are added in user address space, which is separate from kernel space, and thus the kernel doesn't need to be updated. On the other hand, the complete kernel must be updated if a new service is used in a monolithic kernel.

Q4) Suggest whether centralized systems have concurrency transparency property automatically?

- Centralized systems are systems that use client/server architecture where one or more client nodes are directly connected to a central server. This is the most commonly used type of system in many organizations where a client sends a request to a company server and receives the response.
- In these kind of systems multiple clients can access the same resource hence there is no concurrency transparency automatically. We need to separately implement concurrency transparency mechanisms like Time-Stamping

Q5) Explain concept of parallelism transparency

- In principle the DS is supposed to appear to the user as a traditional, uniprocessor timesharing system. But usually the programmer know that systems can contain multiple processors are able to perform many tasks in parallel.
- Unfortunately, current state-of-affairs is nowhere near the concept of this transparency. Programmers who want to use multiple CPU's for a single problem will have to program this explicitly at-least for

the foreseeable future. When this transparency has been achieved it will be time to move on to new fields

Q6) An experimental file server is up 3/4th of the time and down 1/4th of the time due to bugs. How many times this server has to be replicated to get at-least 99% availability.

If the server replicated 2 time

- P(failure) = (1/4)*(1/4) = 0.625 = 6.25%
- Availability = 100-6.25% ~ 93%

If server replicated 3 times

- P(failure) = (1/4)3 = 0.0156 = 1.56%
- Availability = 100-1.56 = 98.44 %

If server replicated 4 times

- P(failure) = (1/4)4 = 0.0039 = 0.39%
- Availability = 100-0.39 = 99.61%

Hence the server must be replicated 4 times

Q7) Suppose a large source program consists of m files to compile. The compilation is to take place on a system with n processes where n>>m. What factors might cause speed-up to be less than the max speed-up where you have m-fold speed-up over single process.

One of the most important performance features of distributed applications is speedup. It's commonly described as the ratio of an application's execution time on a single processor to the same workload's execution time on a system with N processors.

The number of processors and their performance, the connection between the processors, the algorithm used for workload distribution, and other factors all influence speed. When evaluating the speedup of a distributed application, some of these aspects may be difficult to account for.