Introduction system in term of set of applications.

IntroductionAn operating system is a special type of software designed for the purpose of controlling and managing the execution of applications in the computer and acts as an interface between application programs and computer hardware. Generally, the end user sees the computer system in term of set of applications. User is unaware of computer hardware details. A distributed operating system looks at its users as a general centralized operating system but runs on many independent connected systems. Distributed system uses multiple central processors to serve multiple Real-Time applications and multiple users.

Data processing jobs are distributed among the processors accordingly. Processors communicate with each other through various lines (such as high-speed buses or telephone lines). Distributed systems do not share main memory or disks. A distributed operating system is basically a collection of processors connected to a communication network in which each processor has its own local memory and peripheral devices and passing through the communication network between the system’s processor communications happens. The distributed operating system provides a mechanism in which the master operating system splits the work into different slaves and the processor collects the results of all slaves and shows the final result. Distributed operating system is an extension of the network operating system, where distributed applications are running on many computers connected by communication. Distributed operating systems are designed in a manner that does not have complete information about the system environment. Resources in distributed systems differ physically, there is no normal clock between multiple processors, delivery of messages is delayed and messages can be lost too.

Despite all these complexities and difficulties, the distributed operating system should be designed in a way that the user could be able to see a distributed system as a virtual centralized system that is flexible, reliable, secure and easy to use. In order to fulfill this challenge, the designer of distributed operating systems should deal with many design issues. Some of them are as follows: Transparencyaspects in distributed operating systemThe distributed system is considered as the single system bythe user or application programmer.

The user should know where resources andfiles are located and should be transparent to transfer a remote from a localmachine. Transparency makes the system simple and effective. There aredifferent types of transparency which is to combine the distributed system. Transparency in access control where the user is not required or not able to understand whether the resource (hardware or software) is remote or local Distributed operating systems allow users to access remote resources in the same way as local resources. Transparency in location control specifies in two aspects as name transparency and user mobility.

The name transparency will identify the resource (hardware or software) name and any sign should not appear as the physical location of the resource. The resources which are taken from one node to the other should be allowed to move without changing their name. User mobility is related to the fact that no matter which machine a user is logged onto he or she should be able to access a resource with the same name. That is the user should not need to use different names to access similar resources from two separate nodes of the system. Replication transparency states that replicas (extra copies) of files and other resources are created for better performance and reliability of systems on different nodes of distributed operating systems. Users of distributed operating systems should be unaware of the replication of data. It is the responsibility of the distributed system to handle replication issues, which are naming of replicas and replication control.

The system should provide an appropriate name to the replicas and the decision to control the replication such as how many copies of resources should be made or removed should be done entirely by the system automatically in a user transparent manner. Failure transparency in distributed system shows the extent to which errors and failures and later recoveries of hosts and services within the system are invisible to users and applications. If the files are replicated, the partial failure of the system can be handled by providing resources from other nodes. Full failure transparency cannot be achieved by a distributed system, because all types of failures cannot be controlled in a user transparent manner. Migration transparency indistributed operating system allows the user to be unaware of the movement ofinformation or processes within a system without affecting the operations ofthe users and the applications that are running.

In distributed system anobject is migrated from one node to another for better performance, reliabilityand security reasons. Migration decisions are automatically taken by the systemsuch as what object should be moved from where. The distributed system allows its users to use system resources simultaneously. Users can use their resources without interfering between each other, does not reduce the performance of the collateral system Categorizationof transparency based on system goals: System goals Transparency Efficiency Concurrency Parallelism Performance Flexibility Access Location Migration Consistency Access Replication Performance Robustness Failure Replication Reliabilityof distributed operating system with fault handling mechanismDistributed operating systems that handle resources should be prepared properly to increase the reliability of the system. The mistake is a mechanical or algorithmic fault that can cause an error. The fault in the system causes failure of the system. Providing high reliability distributed operating systems should be designed in a way to avoid defects, tolerate defects and detect and recover from defects. Distributed operating systems should be designed in such a way that the occurrence of a mistake can be minimized.

Fault avoidance is most effective when there are only fewer failure modes. Sometimes in the distributed operating system, the fault avoidance mechanism could not be achieved properly. Fault tolerance is the ability of a system running in the case of partial system failure. To handle the mistake is to use the basic technique, redundancy technique in which replicas of resources (additional copies) are made so that one can continue processing after failure of one of them. Replication of resources can help in partial failure, but this disc enhances excessive use and all copies of a file must be consistent. Therefore, the reliability of distributed operating systems and the degree of spent overhead should be designed to maintain a reasonable balance. A system is called k-faulttolerant, if it can continue to work in the event of failure of k components Cristian1991, Nelson 1990. Therefore, it is necessary to continue the “ k +1” replication when the “ k components” of the system fails.

If” k-replicas” are lost due to failure, the remaining one replicationcan be used for continuous functioning of the system. Fault detection andrecovery method deal with the use of hardware and software mechanism todetermine the occurrence of failure and then to correct the system to a stateacceptable for continued operation. Distributedoperating system FlexibilityFlexibility is another most important issue of distributed operating system design. Distributed system design should be flexible due to two reasons – first the distributed operating system should be easily modified and the second system should be easily enhanced.

The most important design factor affecting the flexibility of the distributed operating system, its model is used to design kernel. Distributed operating systems have two commonly used models for the kernel design, which have unbroken Kernel and Micro Kernel.