Unit 11 Operating Systems in Distributed Processing

Structure:

11.1 Introduction

In the last unit, you have studied input / output architecture of the operating systems. Earlier were the days of centralized computing. With the advent of micro and mini computers, distributed processing is becoming more and more popular. Merely having a large central computer with a number of remote terminals connected to it or with a number of computers at different locations with no connection among them do not constitute a distributed processing because neither processing nor data is distributed in any sense. In this unit, we shall discuss operating systems in distributed processing.

Objectives:

After studying this unit, you should be able to:

- explain the characteristics of centralized and distributed processing
- discuss Network Operating System (NOS) architecture
- describe the functions of NOS
- explain Global Operating System (GOS)
- discuss Remote Procedure Call (RPC)
- list out the features of distributed file management

11.2 Centralized and Distributed Processing

Operating systems have moved from single process systems to single processor, multi-user, and multitasking systems. Today the trend is towards

multiprocessor, multitasking systems. Distributed processing and parallel processing are two technologies used to harness the power of a multiprocessor system. A proper mix of the technologies may yield better results.

Distributed processing and parallel processing have a common goal – high throughput using more processors. Then why not use a faster processor? It is difficult to achieve higher throughput out of hardware just by increasing speed of the processor. Moreover faster processors mean high costs. Higher throughput was envisaged by using the available microprocessors and interconnecting them. This is called distributed processing or loosely coupled system. In parallel processing or tightly coupled systems there is only one computer with multiple CPUs. The operating system here is responsible for load distribution, communication and co-ordination.

In distributed processing, computers have to be connected to one another by links enabling electronic data transfer and data sharing among the various connected computers. In a distributed client-server computing environment, the server is huge and handles large databases / computational requests. Clients have smaller processing capability and are spread across different locations. The operating system in such a case has to be restructured to cater to this form of distributed processing. Two approaches to the problem are:

- Network operating system (NOS)
- Global operating system (GOS)

Characteristics of Distributed Processing

Following are the characteristics of distributed processing:

- Processing may be distributed by location.
- Processing is divided among different processors depending on the type of processing done. For example, I/O handled by one processor, user interaction by another and so on.
- Processes can be executing on dissimilar processors.
- Operating system running on each processor may be different.

Characteristics of Parallel Processing

Following are the characteristics of parallel processing:

- All processors are tightly coupled, use shared memory for communication and are present in one case.

- Any processor can execute any job. All processors are similar.
- All processors run a common operating system.

Distributed processing implies a number of computers connected together to form a network. This connection enables distributed applications, data, control or a combination of all of them as against centralized applications, data and control in centralized systems.

Distributed Applications

Distributed applications mean different programs on different computers. This scheme allows the possibility of data capture at the place of its origin. Connections between these computers then allow this data to be shared. Programs / applications could be distributed in two ways. They are:

- Horizontal distribution
- Vertical / hierarchical distribution

In horizontal distribution all computers are at the same level implying that all the computers are capable of handling any functionality. Examples include office automation and reservation systems where many computers in a network are able to reserve, cancel or enquire. Application with all its programs is duplicated at almost all the computers.

In vertical or hierarchical distribution, functionality is distributed among various levels. These levels usually reflect some hierarchical levels in the organization. Computers at each of these levels perform specialized functions. For example, computers at branch level carry out branch level functions and those at zonal level are used for zonal level functions in a banking organization. Computers at each level can be networked together to avail shared data. There are possibilities of connections between levels to enable exchange of data and information. Here applications running on different computers may be the same but for an application program different capabilities may be present at different levels. For example, sales analysis at branch level and sales analysis at zonal level may generate summaries in different formats.

Distribution of Data

In a distributed environment, data can also be distributed similar to distribution of programs. Data for applications could be maintained as:

- Centralized data

- Replicated data
- Partitioned data

In centralized data, data resides only at one central computer that can be accessed or shared by all other computers in the network. For example, master database. This central computer must run an operating system that implements functions of information management. It must keep track of users and their files and handle data sharing, protection, disk space allocation and other related issues. It must also run a front-end software for receiving requests / queries from other computers for data. These requests are then serviced one by one. It is because of this software that this central computer is called a server. Computers connected to the server can have their own local data but shared data has to necessarily reside in the server. In a distributed environment, part of the master database could be centralized and the rest distributed among the connecting computers.

Sometimes a particular database is required very often at each computer in the network. If it is stored only in a central computer, as above, transmitting it from the server to local computers when required is time consuming and an unwanted exercise because the current state of the database may not have changed from a previous state. In such cases, the specific database can be replicated or duplicated in the computer where it is needed often. But to maintain data coherence when part of the database has been updated, the modifications have to be reflected in all the places where it has been duplicated. For example, information about train timings and fares would need replication because this information is needed at all terminals which cater to train bookings / reservations / enquires, the reason being frequency of changes to this particular database is very low.

Data could be distributed in a partitioned way. The entire database is sliced into many parts. Each part of the database then resides on a computer. Processing depends upon the kind of data distribution. Any other computer wanting to access information / data present not locally but at a remote site must send a query and receive the contents needed. If such is the case then each computer will run front-end software to receive queries and act a server for the data stored in it.

Distribution of Control

Control in a distributed environment refers to deciding which program should be scheduled to run next, at which node / computer, what is its data requirement, is there a necessity for data at remote site to be transferred to the node and so on. Network management routines continuously monitor lines and nodes. They help in fault detection and suggest and implement necessary actions to be taken.

Self Assessment Questions

1. Distributed processing and parallel processing are two technologies used to harness the power of a multiprocessor system. (True / False)
2. NOS stands for _____.
3. In _____ distribution all computers are at the same level implying that all the computers are capable of handling any functionality. (Pick the right option)
   a) Horizontal
   b) Vertical
   c) Parallel
   d) Hierarchical

11.3 Network Operating System (NOS) Architecture

The architecture of typical NOS is shown in Figure 11.1. The basic features in any NOS are explained by tracing the steps involved in a remote read. It is assumed that shared data resides on the server and clients are those computers in the network (other than the server) that want to access the shared data.

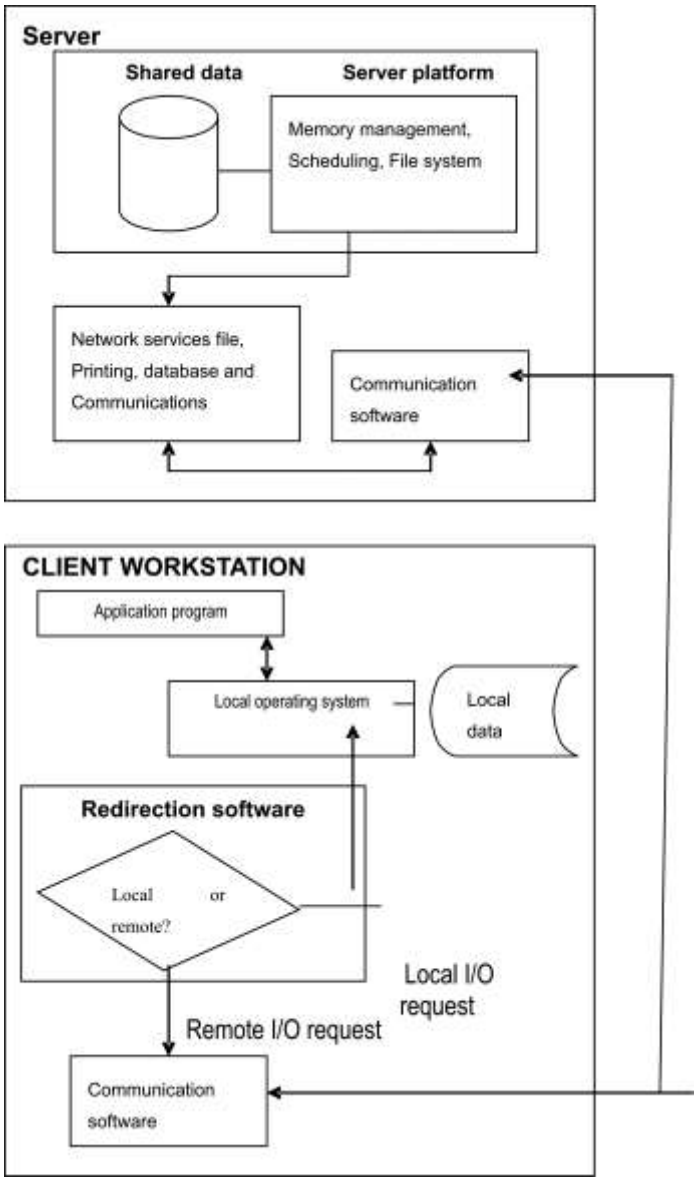

Fig. 11.1: NOS architecture

Let's see the features of NOS architecture:

- Software called redirection software exists in the client.
- A system call generated by an application program not related to any I/O function is handled by the local operating system (LOS) running on the client.

- In a non-NOS environment all I/O by an application program is to the LOS only. However, in the case of NOS environment this cannot be assumed. I/O may be to a local database or a remote database. In such a case a call is made to the redirection software of the NOS. The application program making this I/O call has knowledge about the location of the data (local / remote) and hence requests either the LOS for local data or the NOS for shared data. The NOS differentiates between a LOS I/O call and a NOS I/O call.
- If the request is for remote data then the call has to be processed as a remote procedure call (RPC) from the client to the server. In response to this request, data traverses back to the client from the server. Communication management software handles the request for data and the actual data. This software resides both on the server as well as the client and ensures that a message is communicated between client and the server without any error and implements network functions such as packetizing, routing, error and flow control.
- For a remote request the redirection software on the client sends a request to the communication management software on the client.
- The communication management software on the client generates a RPC and sends it across the network.
- The communication management software on the server receives the request and in turn requests the network services software on the server itself for the client's request. This software is responsible for sharable resources such as files, disks, databases and printers. The software receives many such requests from different clients, generates a task for each one of them and schedules them for service. Thus NOS implements some kind of multitasking to service multiple tasks. Since network services software accesses shared resources, access control and protection are implemented.
- The network services software on the server communicates with the information management module of the operating system running on the server to get the requested data. Two approaches are possible. In one approach, capabilities of information management are built into the NOS such as in NetWare. In the other approach, a separate operating system such as UNIX runs on the server and the network services software

module of the NOS generates calls to the operating system, in this case, UNIX running on the server for required data.

- The network services software on the server sends the required data to the communication management software on the server to be sent to the client.
- The communication management software on the server also implements network functions such as packetizing, routing, sequence control, error and flow control to ensure error free data transfer to the client.
- The communication management software on the client now sends the received data to the application program so that it proceeds.

NOSs are available on LANs. LAN is an interconnection of a number of workstations to form a network. The network also has a large and more powerful computer attached to it. This computer called the server has a large disk and a printer attached to it. The server stores data that can be accessed by clients connected to the network. The clients in the form of workstations have small local memories that can be used for storing frequently accessed data once accessed from the server. Workstations can also be diskless in which case they have no local memory. The LOS is also downloaded into main memory during power up. All data in this case is requested and got from the server.

Self-Assessment Questions

4. In NOS, the shared data resides on the server and clients access the shared data. (True / False)
5. RPC stands for _____.
6. The _____ software on the client generates a RPC and sends it across the network. (Pick the right option)
   a) Network Management
   b) Communication Management
   c) Redirection
   d) File / Printer Service

11.4 Functions of NOS

The main functions of NOS can be summarized as follows:

- Redirection

- Communication management
- File / printer services
- Network management

Redirection

Redirection software normally resides on the client. It may reside on the server also; because, if it is not a dedicated one then user of the server machine may want access to other computers. When does the redirection software actually work? An interrupt is executed by a system call generated, say for an I/O. It is at the time of execution of the interrupt that redirection software intercepts to check if the I/O is local / remote. If it is local, processing continues. If it is remote the redirection software has to generate a request to the server. But generating a request to the server has problems. The operating system running on the server may be different from that on the local machine generating the request. Also system architecture of the server may be different from the client. Therefore some conversion is necessary.

Communication Management

The communication management software runs on both the client and the server. It is responsible for communication management. It is concerned with error-free transmission of messages (requests and data) to the destination. The ordinary operating system depends on separate software for this purpose. But in a NOS environment communication management software is built into the NOS as a part of it. Thus it resides on all clients and the server. It consists of a number of modules corresponding to the Open System Interconnect (OSI) layers.

File / Printer Services

File / printer resources are controlled by these services. This software runs only on the server. Requests for shared resources are queued up, scheduled and then run as separate tasks, thus making the NOS a multitasking operating system.

Network Management Software

Network management software is responsible for monitoring the network and its components such as computers, modems, repeaters, lines, adapters, multiplexers and many more. Special software enables online testing of these equipments from time to time, checks their status and hence monitors

the entire network. The network management software is responsible for all this. It maintains a list of hardware equipment along with its location and status. The list is updated when additional equipment is added or when equipment is down for repair. It generates reports based on which action can be taken in terms of repair / replacements. It helps routing algorithms to route data on appropriate paths. The network management software resides on top of the existing operating system in ordinary operating systems. But in a NOS environment it is part of the NOS.

Self-Assessment Questions

7. File / Printer resources software runs both on client and server. (True / False)
8. OSI stands for _____.
9. _____ software is responsible for monitoring the network and its components such as computers, modems, repeaters, lines, adapters, multiplexers and many more. (Pick the right option)
   a) Network Management
   b) Communication Management
   c) Redirection
   d) File / Printer

11.5 Global Operating System (GOS)

The NOS is responsible for activities such as memory and process management on the server. The NOS converts a request into a task, schedules and executes it. Memory and processing power in all other computers in the network is not tapped to the maximum by NOS. This is exactly what the GOS attempts to do. It has a list of processes executing on different machines and the resources needed by each one of them. Relatively free processors can be scheduled with tasks for execution. Memory is managed at a global level. The various functions of the GOS are:

- User interface
- Information management
- Process / object management
- Memory management
- Communication management
- Network management

A typical GOS environment is depicted in the figure 11.2. Part of the kernel of a GOS is duplicated at all sites. This kernel contains software to control hardware. Resources like information, memory, etc are managed by software that need not be replicated.

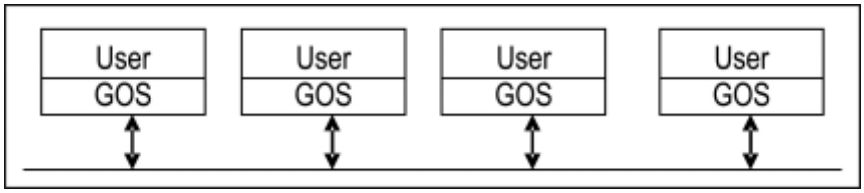

Fig. 11.2: GOS environment

Migration

The GOS has a pool of resources that it allocates to various processes / tasks at a global level. Migrations are necessary for optimal use of available resources. Migrations include:

- Data migration
- Computation migration
- Process migration

Data migration involves movement of data. A program running at a site X wants access to a file at site Y. Two options exist:

- Send the full file from Y to X
- Send only required portion of the file from Y to X

The first option is similar to the approach of a file server whereas the second is similar to a database server. Software for sending the full file is simple. But the network will be loaded and in case the file is updated at site X, the entire file has to be again sent back to Y. If only required portions of a file are sent then network load is less but software to handle this is complex. Depending on requests for remote data, the GOS may migrate portion of data from one node to another or may replicate data to improve performance. This also brings with it the problems of data integrity.

The GOS may sometimes resort to computation migration. If nodes are distributed in a hierarchical fashion then data migration will need to transfer all files between levels. Alternatively, if computation migration is followed then a process on one node can request for execution of another process at a remote site through a RPC. The results of this computation at remote site are then sent back for use. Here data file transfer is avoided.

Sometimes a process may be scheduled on a node that does not have the necessary requirements for the process because of which the process does not complete execution but is waiting in a blocked state for a long time. Since it was the only processor at the time of allocation it runs the process. Now that another processor with higher capacity is free, the GOS should be able to migrate the process to the new processor. There exists a tradeoff between the gain in performance of the migrated process and the overheads involved.

GOS may resort to process migration to enforce:

- Load balancing: to have a uniform utilization of available resources
- Special facilities: to use hardware / software facilities available at a particular node
- Reducing network load: process execution at a proper node reduces data migration and hence the load on the network.

Resource Allocation/ De-allocation

The GOS maintains a global list of all resources and allocates them to processes. This also includes migrated processes. The resource allocation may lead to deadlocks. Deadlock handling in distributed systems is complex due to difficulties in maintaining an updated list of global resources. There is also a possibility of a false deadlock alarm. This may be caused because of incorrect information about resources that in turn may be due to delay in resource status reaching the global list. Deadlock detection can be centralized or a distributed functions. Deadlocks can also occur in the communication system due to buffers getting full.

Self Assessment Questions

10. The NOS is not responsible for activities such as memory and process management on the server. (True / False)
11. GOS stands for _____.
12. Migrations are necessary for optimal use of available resources. Migrations include: (Pick the right option)
    a) Data migration
    b) Computation migration
    c) Process migration
    d) All of the above

11.6 Remote Procedure Call (RPC)

A distributed environment consists of servers and clients. Server is a computer that offers services of shared resources. Client is a computer that requests for a shared resource present on the server through a request. A procedure is present on the server to locate and retrieve data present on a shared device attached to it. This procedure is part of the operating system running on the server. When a client requests for some data on the server this procedure on the server operating system is called remotely from the client. Hence it is called a remote procedure call (RPC).

Message Passing Schemes

RPC can be considered as a special case of a generalized remote message-passing scheme as shown in figure 11.3. The message handling module forms the interface that runs on all the nodes connected in the network. It interfaces with processes running on the nodes using primitives like SEND and RECEIVE. These modules handle communication across the network. Communication management functions are executed to ensure error-free communication.

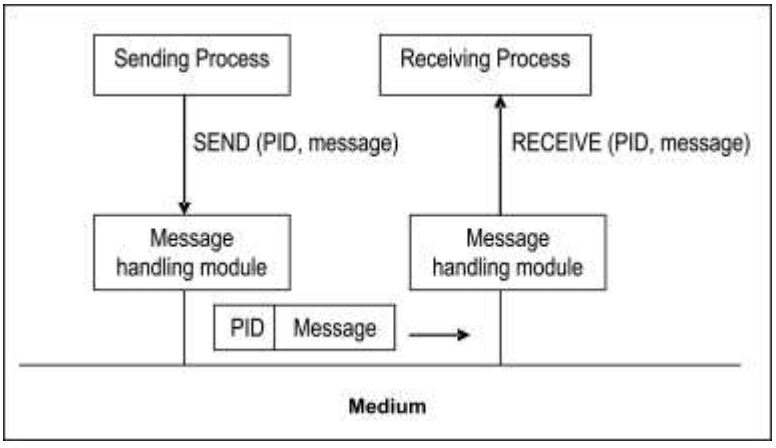

Fig. 11.3: Message passing scheme

Types of Services

Message passing can be of two types. They are:

- Reliable service
- Unreliable service

A virtual circuit analogous to a telephone service is an example of a reliable service whereas a datagram analogous to the postal service is an example for unreliable services. A reliable service ensures that the receiver receives the message sent by a sender correctly and properly in sequence. The overhead in this service includes an increased load on the network. An unreliable service only guarantees a high probability that a sent message is correctly received in proper order.

Message passing schemes could also be categorized as:

- Blocking
- Non-blocking

In the blocking scheme, the process on the client that has requested for service from the server gets blocked until it receives back the data, whereas in the non-blocking scheme, the process requesting for service continues without waiting.

RPC can be viewed as an enhancement of a reliable blocking message-passing scheme to execute a remote procedure on another node. The message in this case is not a general one but specifies the procedure to be executed on the remote node along with required parameters.

Calling Procedure

A general format for an RPC could be as follows:

CALL P (A, B)

where, P is the called procedure
A is the passed parameters
B is the returned parameters

Parameters can be passed either by value or by reference. When parameters are passed by value, the actual parameters are passed. Thus A and B will be actual parameters. If parameters are passed by reference then the addresses of the actual parameters are passed.

In RPC call by reference is very difficult because it is difficult to let processors on different machines to share a common address space. Hence call by reference does not make sense in RPC. It becomes tedious and time consuming. It also increases the load on the network. That is why only call by value method is used in RPC. A general schematic of RPC is shown in the figure 11.4. The client process issues an RPC and gets blocked. The

interface process completes the call and returns the results after which the client process becomes ready again.

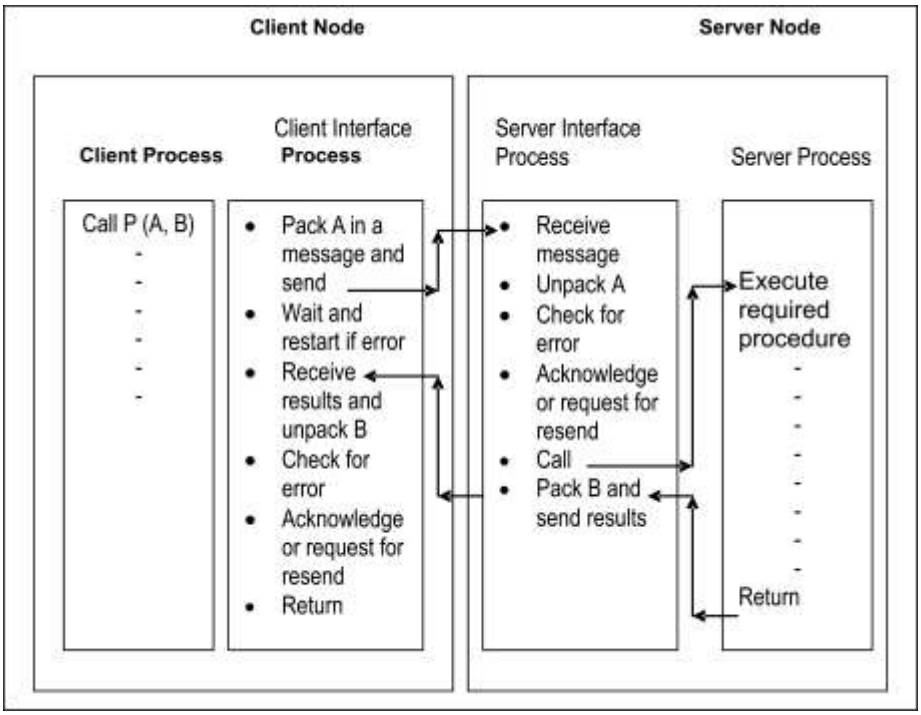

Fig. 11.4: A general schematic of RPC

Parameter Representation

If an RPC is issued between processes running on identical machines with same operating systems then parameters passed will be identical for a given language. But this is not the case if the machine architecture or the operating system or the programming language differs. One approach to this problem could be to have a common standard format. Then each interface module will have routines to convert from / to its own formats to / from the standard format. These routines will have to be present in all nodes as well as the server.

Ports

If a server provides multiple services then normally a port number is associated with each service. For example, port number 1154 for listing current users, port number 2193 for opening a file and so on. RPC makes

use of these port numbers. This simplifies communication. Hence a message sent as a RPC to a remote node contains among other information the port number and parameters for the service. The interface module on the remote node reads the port number and then executes the appropriate service.

11.7 Distributed File Management

A network has many nodes. Each node has files in its local database. In NOS a user has to specify the exact location of a file to get it transferred to his / her node. But this is not required in GOS.

Sometimes in a NOS environment it is advantageous to keep multiple copies of the same file at different nodes. This reduces transfer time and also traffic on the network. The nearest node having the file can then satisfy a user request. To implement this, the node requesting the file, the remote node where the file is present and the frequency of requests need to be known. This is a dynamic situation since the pattern for file requests change with time. Hence the number of nodes to replicate a file is a dynamic issue. Maintaining data integrity is a problem as will have to be made at multiple locations.

Each node in the network runs its own local operating system and thus has its own file system. This local file system (LFS) is responsible for allocating space to a file, maintaining buffers, tables like FAT and so on. Services for file creation, deletion, read and write are provided by it. It maintains the directory structure and associated files. The functions of the LFS on a remote file are carried out by the distributed file system (DFS). It allows the users to see an entire structure of files and directories present in all the nodes put together as a hierarchy. An important implementation consideration in the design of DFS is the policy to be used to implement file operations, especially write and update operations. DFS has to have software to interface with the operating system running on different nodes. This software should be present on all the nodes. If all nodes run the same operating system then complexity of DFS is greatly reduced.

UNIX has a feature called RFS that is a DFS for UNIX. SUN has its NFS that is again a DFS and is part of the SunOS operating system. NetWare-

386 can support multiple machines and multiple networks / distributed file systems at the same time.

Self Assessment Questions

13. A procedure is present on the server to locate and retrieve data present on a shared device attached to it. (True / False)
14. In RPC _____ is very difficult because it is difficult to let processors on different machines to share a common address space.
15. _____ can support multiple machines and multiple networks/ distributed file systems at the same time. (Pick the right option)
    a) DOS
    b) NetWare-386
    c) Windows
    d) Linux

11.8 Summary

Let's recapitulate important points discussed in this unit:

- Operating systems have moved from single process systems to single processor, multi-user, and multitasking systems.
- Distributed processing and parallel processing have a common goal i.e. high throughput using more processors.
- In distributed processing, computers have to be connected to one another by links enabling electronic data transfer and data sharing among the various connected computers.
- The main functions of NOS include: Redirection, Communication management, File / printer services and Network management.
- The GOS has a pool of resources that it allocates to various processes / tasks at a global level.

11.9 Terminal Questions

1. Distinguish between centralized processing and distributed processing.
2. Explain Network Operating System (NOS) architecture in detail.
3. What is Global Operating System? Explain.
4. Explain the execution of a RPC.

11.10 Answers

Self Assessment Questions

1. True
2. Network Operating System
3. a) Horizontal
4. True
5. Remote Procedure Call
6. b) Communication Management
7. False
8. Open System Interconnect
9. a) Network Management
10. False
11. Global Operating System
12. d) All of the above
13. True
14. Call by reference
15. b) Netware-386

Terminal Questions

1. In distributed processing, computers have to be connected to one another by links enabling electronic data transfer and data sharing among the various connected computers. (Refer section 11.2)
2. The basic features in any NOS are explained by tracing the steps involved in a remote read. It is assumed that shared data resides on the server and clients are those computers in the network (other than the server) that want to access the shared data. (Refer section 11.3)
3. Part of the kernel of a GOS is duplicated at all sites. This kernel contains software to control hardware. Resources like information, memory, etc. are managed by software that need not be replicated. (Refer section 11.5)
4. A procedure is present on the server to locate and retrieve data present on a shared device attached to it. This procedure is part of the operating system running on the server. When a client requests for some data on the server this procedure on the server operating system is called remotely from the client. Hence it is called a remote procedure call (RPC). (Refer section 11.6)