

# MCS 022 2016-2017 session Ignou Study Helper

[www.ignouite.blogspot.com](http://www.ignouite.blogspot.com)

Q.1. a) Differentiate between Microkernel architecture and Kernel architecture?

Ans1 (a) Kernel Architecture:-

The kernel is the core of an operating system. It is the software responsible for running programs and providing secure access to the machine's hardware. Since there are many programs, and resources are limited, the kernel also decides when and how long a program should run. This is called scheduling. Accessing the hardware directly can be very complex, since there are many different hardware designs for the same type of component. Kernels usually implement some level of hardware abstraction (a set of instructions universal to all devices of a certain type) to hide the underlying complexity from applications and provide a clean and uniform interface. This helps application programmers to develop programs without having to know how to program for specific devices. The kernel relies upon software drivers that translate the generic command into instructions specific to that device.

An operating system kernel is not strictly needed to run a computer. Programs can be directly loaded and executed on the "bare metal" machine, provided that the authors of those programs are willing to do without any hardware abstraction or operating system support. This was the normal operating method of many early computers, which were reset and reloaded between the running of different programs. Eventually, small ancillary programs such as program loaders and debuggers were typically left in-core between runs, or loaded from read-only memory. As these were developed, they formed the basis of what became early operating system kernels. The "bare metal" approach is still used today on many video game consoles and embedded systems, but in general, newer systems use kernels and operating systems.

Four broad categories of kernels:

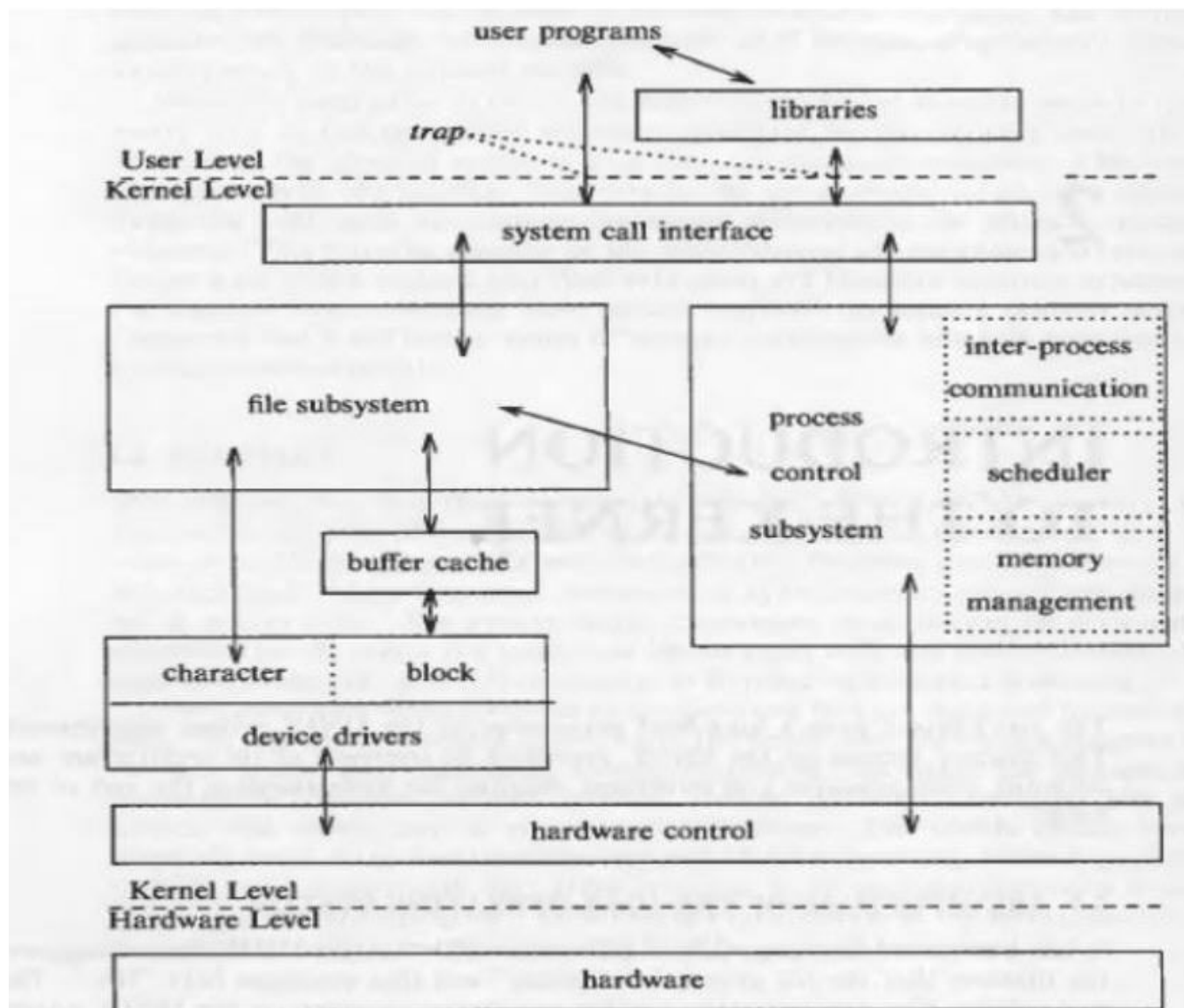
*Monolithic Kernel* provide rich and powerful abstractions of the underlying hardware.

*MicroKernels* provide a small set of simple hardware abstractions and use applications called servers to provide more functionality.

*ExoKernels* provide minimal abstractions, allowing low-level hardware access. In exokernel systems, library operating systems provide the abstractions typically present in monolithic kernels.

*Hybrid (modified microkernels)* are much like pure microkernels, except that they include some additional code in kernelspace to increase performance.

A block diagram of the kernel, showing various modules and their relationships to each other. In particular, it shows the file subsystem on the left and the process control subsystem on the right, the two major component of the kernel.



**Figure 2.1. Block Diagram of the System Kernel**

Assembly language programs may invoke system calls directly without a system call library, however. Programs frequently use other libraries such as the standard I/O library to provide a more sophisticated use of the system calls. The libraries are linked with the programs at compile time.

MICROKERNAL ARCHITECTURE-----

In computer science, a microkernel (also known as  $\mu$ -kernel) is the near-minimum amount of Software that can provide the mechanisms needed to implement an operating system (OS). These mechanisms include low-level address space management, thread management, and inter process communication (IPC).

If the hardware provides multiple rings or CPU modes , the microkernel may be the only software executing at the most privileged level, which is generally referred to as supervisor or Kernal Mode. Traditional operating system functions, such as device drivers, protocol stacks and file system, are typically removed from the microkernel itself and are instead run in user space.

This structures the operating system by removing all nonessential portions of the kernel and implementing them as system and user level programs.

Generally they provide minimal process and memory management, and a communications facility.

Communication between components of the OS is provided by message passing.

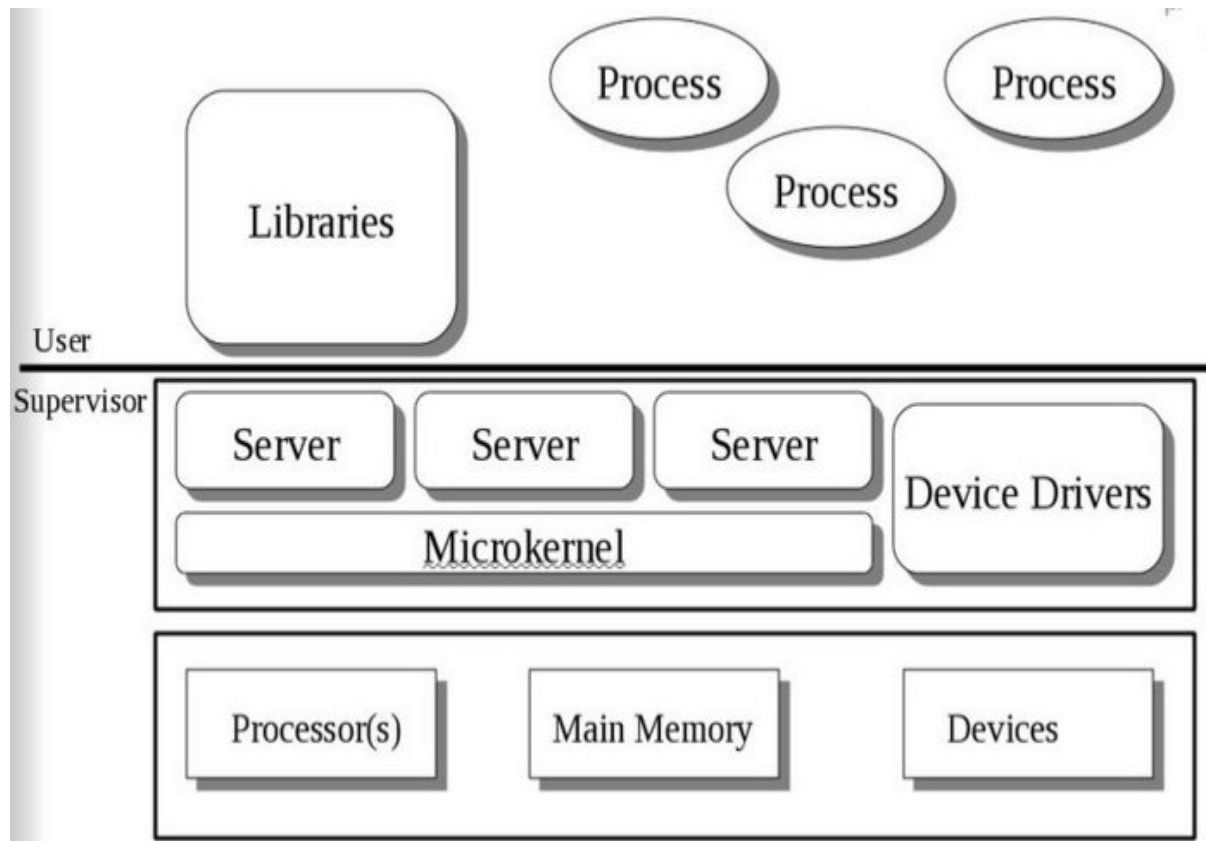
The *benefits* of the microkernel are as follows:

Extending the operating system becomes much easier.

Any changes to the kernel tend to be fewer, since the kernel is smaller.

The microkernel also provides more security and reliability.

Main *disadvantage* is poor performance due to increased system overhead from message passing.



Q.1.(b) Define Distributed File System? Explain how it is implemented in Windows 2000?

Ans.1 (b) Distributed File System:-

The Distributed File System (DFS) allows files and directories in various places to be combined into one directory tree. Only Windows 2000 Servers can contain DFS root directories and they can have only one.

### DFS Characteristics

The permissions of shared folders that are part of the DFS are still the same.

Shares with important information can be replicated to several servers providing fault tolerance.

The DFS root must be created first.

### DFS Components

DFS root - A shared directory that can contain other shared directories, files, DFS links, and other DFS roots. One root is allowed per server. Types of DFS roots:

Stand alone DFS root - Not published in Active Directory, cannot be replicated, and can be on

any Windows 2000 Server. This provides no fault tolerance with the DFS topology stored on one computer. A DFS can be accessed using the following syntax:

`\\Server\DFSname`

Domain DFS root - It is published in Active Directory, can be replicated, and can be on any Windows 2000 Server. Files and directories must be manually replicated to other servers or Windows 2000 must be configured to replicate files and directories. Configure the domain DFS root, then the replicas when configuring automatic replication. Links are automatically replicated. There may be up to 31 replicas. Domain DFS root directories can be accessed using the following syntax:

`\\domain\DFSname`

DFS link - A pointer to another shared directory. There can be up to 1000 DFS links for a DFS root.

DFS administration is done on the Administrative Tool, "Distributed File System". This tool is on all Windows 2000 Server computers, and Windows 2000 Professional computers that have the ADMINPAK installed.

## Client Computers

Windows 2000 Server

Windows 2000 Professional

Windows NT 4.0 or later Server and Workstation

Windows 95 and Windows 98 with DFS client software. (No access to DFS links on NetWare servers).

## Replication

The File Replication Service (FRS) can be used to replicate DFS shares automatically.

Implemented of DFS in Windows 2000 :-

Windows 2000 Server is used primarily for web, application, print and file servers.

## Hardware Support

Up to 4 Gigabytes of RAM.

Up to 4 microprocessors.

## Features not provided by Windows 2000 Professional

Windows 2000 server can be a domain controller with the ability to have a read/write copy of Active Directory data.

Disk Quotas - Disk space use is tracked for each user.

DFS - Distributed file system support. Shares that are stored on various remote computers can appear as one share

Supported Servers:

Internet Information Server

SQL Server

Exchange Server

Systems Management Server

RADIUS - Remote Authentication Dial-In User Service

SNA Server

### Supported Protocols

Network Protocols

IP

IPX

AppleTalk

Routing Protocols

RIP version 2 (Routing Information Protocol)

OSPF - Open shortest path first.

ATM - Asynchronous Transfer Mode.

Q.2.(a) Explain the differences between the following groups of Windows 2000 operating system:

(i) Global groups

(ii) Domain Local groups

(iii) Local groups

(iv) System groups

## Ans.2.(a)

A group is a collection of user and computer accounts, contacts, and other groups that you can manage as a single unit. Users and computers that belong to a particular group are referred to as group members.

Groups in Active Directory Domain Services (AD DS) are directory objects that reside in a domain and in organizational unit (OU) container objects. AD DS provides a set of default groups at installation. It also provides an option to create groups.

You can use groups in AD DS to:

Simplify administration by assigning permissions on a shared resource to a group, rather than to individual users. Assigning permissions to a group assigns the same access to the resource to all members of that group.

Delegate administration by assigning user rights once to a group through Group Policy. You can then add members to the group that you want to have the same rights as the group.

Create e-mail distribution lists.

### (i) Global Groups:-

Members of global groups can include accounts from the same domain as the parent global group and global groups from the same domain as the parent global group. Members of these groups can be assigned permissions in any domain in the forest.

Use groups with global scope to manage directory objects that require daily maintenance, such as user and computer accounts. Because groups with global scope are not replicated outside their own domain, you can change accounts in a group having global scope frequently without generating replication traffic to the global catalog.

Although rights and permissions assignments are valid only within the domain in which they are assigned, by applying groups with global scope uniformly across the appropriate domains, you can consolidate references to accounts with similar purposes. This simplifies and rationalizes group management across domains. For example, in a network with two domains, Europe and UnitedStates, if there is a group with global scope called GLAccounting in the UnitedStates domain, there should also be a group called GLAccounting in the Europe domain (unless the accounting function does not exist in the Europe domain).

### (ii) Domain Local Group:-

Members of domain local groups can include other groups and accounts from Windows NT, Windows 2000, Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2 domains. Members of these groups can be assigned permissions only within a domain.

Groups with domain local scope help you define and manage access to resources within a single domain. These groups can have the following as their members:

Accounts from any domain

Global groups from any domain

Universal groups from any domain

Domain local groups, but only from the same domain as the parent domain local group

A mixture of any of the above

For example, to give five users access to a particular printer, you can add all five user accounts in the printer permissions list. If, however, you later want to give the five users access to a new printer, you again have to specify all five accounts in the permissions list for the new printer.

With a little planning, you can simplify this routine administrative task by creating a group with domain local scope and assigning it permission to access the printer. Put the five user accounts in a group with global scope, and add this group to the group that has domain local scope. When you want to give the five users access to a new printer, assign the group with domain local scope permission to access the new printer. All members of the group with global scope automatically receive access to the new printer.

### (iii) Local Groups:-

Local groups, such as the Domain Admins group, are security groups that are created automatically when you create an Active Directory domain. You can use these predefined groups to help control access to shared resources and to delegate specific, domain-wide, administrative roles.

Many default groups are automatically assigned a set of user rights that authorize members of the group to perform specific actions in a domain, such as logging on to a local system or backing up files and folders. For example, a member of the Backup Operators group has the right to perform backup operations for all domain controllers in the domain.

When you add a user to a group, the user receives the following:

All the user rights that are assigned to the group

All the permissions that are assigned to the group on any shared resources

Local groups are located in the Builtin container and the Users container. The local groups in the Builtin container have a group scope of Builtin Local. Their group scope and group type cannot be changed. The Users container contains groups that are defined with global scope and groups that are defined with domain local scope. You can move groups that are located in these containers to other groups or OUs within the domain, but you cannot move them to other domains.

### (iv) System Groups:-

*System groups* can contain members from any Windows 2000 domain in the forest, and can be granted permissions in any domain in the forest or in trusted forests. Though universal groups can have members from mixed mode domains in the same forest, members from such domains do not have the universal group added to their access tokens because universal



groups are not available in mixed mode. Though you can add users to a universal group, it is recommended that you restrict membership to system groups. Note that universal groups are only available in native mode domains.

You can use universal groups to build groups that perform a common function within an enterprise. An example of this is virtual teams. The membership of such teams in a large company could be nation-wide, or world-wide, and almost certainly forest-wide, with team resources being similarly distributed. In these circumstances, universal groups could be used as a container to hold global groups from each subsidiary or department, with the team resources being protected by a single ACE for the universal group.

System groups and their members are listed in the Global Catalog (GC). Though global and domain local groups are also listed in the GC, their members are not. This has implications for GC replication traffic. It is recommended that you use universal groups with care. If your entire network has high-speed connectivity, you can simply use universal groups for all your groups, and benefit from not having to manage global groups and domain local groups. If, however, your network spans wide area networks (WANs), you can improve performance by using global groups and domain local groups.

If you use System groups and domain local groups, you can also designate as universal groups any widely used groups that are seldom changed.

Q.2.(b) Explain the role of NAME SERVERS and RESOLVERS in DNS Architecture?

Ans.2.(b) Domain Name Server(DNS):-

DNS is the abbreviation of Domain Name System which is a stratified naming system for services, computer and for any other possessions attached to the network or internet. DNS (Domain Name System) is responsible to links a variety of information with domain names allocated to each of the participants. The main task of DNS (Domain Name System) is the translation human language into the binary code. It is also serve as the distributed database that offers mapping among Internet Protocol addresses and host names. DNS makes facilitate the client to consign a domain name to groups of internet users in a significant way.

# DNS Architecture

DNS architecture is a hierarchical distributed database and an associated set of protocols that define:

A mechanism for querying and updating the database.

A mechanism for replicating the information in the database among servers.

A schema of the database.

DNS originated in the early days of the Internet when the Internet was a small network established by the United States Department of Defense for research purposes. The host names of the computers in this network were managed through the use of a single HOSTS file located on a centrally administered server. Each site that needed to resolve host names on the

network downloaded this file. As the number of hosts on the Internet grew, the traffic generated by the update process, as well as the size of the HOSTS file, increased. The need for a new system, which would offer features such as scalability, decentralized administration, support for various data types, became more and more obvious.

## Role of Name Server:-

### Name servers

The DNS system is upheld by a dispersed database system. This system employs the client server model and the knobs of database are the name servers. Every domain has minimum one authorized DNS server which distributed information about domain and name server of every domain subsidiary to it.

### Authoritative name server

It is a name server which is responsible to answer about the configuration that it is held by an original source. Only Authoritative name server returned answer to inquiries related to domain name that have been exclusively configured by the administrator. Authoritative name server can be whichever slave master which uses an automatic method of the DNS protocol in contact in the company of its master server which maintains alike copy of the master account. Each zone of DNS ought to be a set of authoritative name servers. After the registration of domain name their setting up has need of the assignment of a primary name server and minimum one secondary server. The necessity of manifold name servers intends to make the domain still practical yet if one name server happen to unreachable or untreatable. The title of a primary name server is exclusively resolute by the precedence specified to the domain name registrar. For this reason usually only the completely competent domain name of the name server is compulsory.

## Role of Resolver:-

### DNS resolvers

The DNS (Domain Name System) client side is called DNS resolver which is responsible for starting and sequencing the inquiries that eventually guide to a complete resolution of the sources required. DNS resolvers and DNS server are performing recursively on behalf of the resolver.

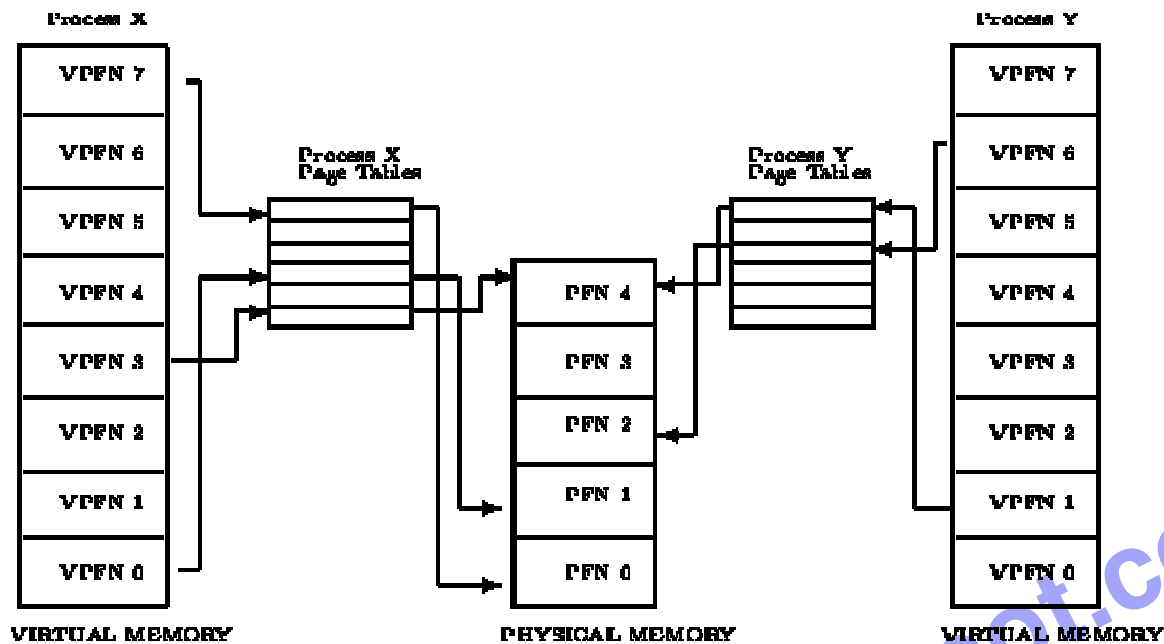
Resolving typically involve iterating through numerous name servers to locate the desirable information. Some resolvers purpose simplistically and can converse only with a particular name server and these all called stub resolvers which are rely on a recursive name server to execute the work of finding information for them.

### DNS related files

DNS (Domain Name System) consist of boot which is a BIND boot configuration file, Cache.dns use for uploading, Root.dns which is a root zone file, and zone\_name.dns which is a used when any zone entered and configured for a server.

Q.3.(a) Explain the abstract model of Virtual Memory used in Linux operating system. Give a suitable diagram to explain its working.

Ans.3.(a) Abstract Model of Virtual Memory In Linux Operating System:-



The virtual PFN,

The physical PFN that it maps to,

Access control information for that page.

In this model, both virtual and physical memory are divided up into handy sized chunks called pages. These pages are all the same size, they need not be but if they were not the system would be very hard to administer. Linux on Alpha AXP uses 8 Kbyte pages. Each of these pages is given a unique number, the Page Frame Number (PFN). \_\_\_ For every instruction in a program, for example to load a register with the contents of a location in memory, the CPU performs a mapping from a virtual address to a physical one. Also, if the instruction itself references memory then a translation is performed for that reference.

The address translation between virtual and physical memory is done by the CPU using page tables which contain all the information that the CPU needs. Typically there is a page table for every process in the system. Figure ---- a simple mapping between virtual addresses and physical addresses using page tables for *Process X* and *Process Y*. This shows that *Process X*'s virtual PFN 0 is mapped into memory in physical PFN 1 and that *Process Y*'s virtual PFN 1 is mapped into physical PFN 4. Each entry in the theoretical page table contains the following information:

As the processor executes a program it reads an instruction from memory and decodes it. In decoding the instruction it may need to fetch or store the contents of a location in memory. The

processor then executes the instruction and moves onto the next instruction in the program. In this way the processor is always accessing memory either to fetch instructions or to fetch and store data.

In a virtual memory system all of these addresses are virtual addresses and not physical addresses. These virtual addresses are converted into physical addresses by the processor based on information held in a set of tables maintained by the operating system.

To make this translation easier, virtual and physical memory are divided into handy sized chunks called *pages*. These pages are all the same size, they need not be but if they were not, the system would be very hard to administer. Linux on Alpha AXP systems uses 8 Kbyte pages and on Intel x86 systems it uses 4 Kbyte pages. Each of these pages is given a unique number; the page frame number (PFN).

In this paged model, a virtual address is composed of two parts; an offset and a virtual page frame number. If the page size is 4 Kbytes, bits 11:0 of the virtual address contain the offset and bits 12 and above are the virtual page frame number. Each time the processor encounters a virtual address it must extract the offset and the virtual page frame number. The processor must translate the virtual page frame number into a physical one and then access the location at the correct offset into that physical page. To do this the processor uses *page tables*.

Q.3.(b) Write a shell script in LINUX to count the number of words in a given file?

Ans.3.(b) Shell Scripting:-

In Linux, shells like bash and korn support programming construct which are saved as scripts. These scripts become shell commands and hence many Linux commands are script.

A system administrator should have a little knowledge about scripting to understand how their servers and applications are started, upgraded, maintained or removed and to understand how a user environment is built.

read file

I=`wc -l \$file|cut -d " " -f 1`

echo \$I

count=0

```
coch=0
```

```
for (( i=1; i<=l; i++))
```

```
do
```

```
    n=1
```

```
    line=`head -$i $file|tail -1`
```

```
    echo $line
```

```
    ch=`echo $line|cut -c $n`
```

```
    echo character is $ch
```

```
    while [ "$ch" != "" ]
```

```
    do
```

```
        if [ "$ch" = " " ]
```

```
        then
```

```
            count=`expr $count + 1`
```

```
        else
```

```
            coch=`expr $coch + 1`
```

```
        fi
```

```
        n=`expr $n + 1`
```

```
        ch=`echo $line|cut -c $n`
```

```
    done
```

```
done
```

```
echo no. of space $count
```

```
echo no. of characters $coch
```

Q.4.(a) Which application layer protocol is used by network management frameworks to manage and monitor network devices? Explain its architecture and working.

Ans.4.(a)

## Network Management Protocols and Features

Proper network management is a critical component of an efficient network. Network administrators need tools to monitor the functionality of the network devices, the connections between them, and the services they provide. SNMP has become the de facto standard for use in network management solutions and is tightly connected with remote monitoring (RMON) and Management Information Bases (MIB). Each managed device in the network has several variables that quantify the state of the device. You can monitor managed devices by reading the values of these variables, and you can control managed devices by writing values into these variables.

This section introduces SNMP and describes the differences between SNMP versions 1, 2, and 3. The role of MIBs in SNMP and RMON monitoring is described, and Cisco's network discovery protocol, Cisco Discovery Protocol (CDP), is introduced. The section concludes with a description of methods for gathering network statistics.

The network management architecture consists of the following:

**Network management system (NMS):** A system that executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources that are required for network management.

**Network management protocol:** A protocol that facilitates the exchange of management information between the NMS and managed devices, including SNMP, MIB, and RMON.

**Managed devices:** A device (such as a router) managed by an NMS.

**Management agents:** Software, on managed devices, that collects and stores management information, including SNMP agents and RMON agents.

**Management information:** Data that is of interest to a device's management, usually stored in MIBs.

A variety of network management applications can be used on a network management system; the choice depends on the network platform (such as the hardware or operating system). The management information resides on network devices; management agents that reside on the device collect and store data in a standardized data definition structure known as the *MIB*.

The network management application uses SNMP or other network management protocols to retrieve the data that the management agents collect. The retrieved data is typically processed and

prepared for display with a GUI, which allows the operator to use a graphical representation of the network to control managed devices and program the network management application.

### Protocols and Standards

Several protocols are used within the network management architecture.

### Key Point

SNMP is the simplest network management protocol. SNMP version 1 (SNMPv1) was extended to SNMP version 2 (SNMPv2) with its variants, which were further extended with SNMP version 3 (SNMPv3).

The MIB is a detailed definition of the information on a network device and is accessible through a network management protocol, such as SNMP.

RMON is an extension of the MIB. The MIB typically provides only static information about the managed device; the RMON agent collects specific groups of statistics for long-term trend analysis.

### SNMP

SNMP has become the de facto standard for network management. SNMP is a simple solution that requires little code to implement, which enables vendors to easily build SNMP agents for their products. In addition, SNMP is often the foundation of the network management architecture. SNMP defines how management information is exchanged between network management applications and management agents.

Q.4.(b) Explain drive mapping facility in Windows 2000 with suitable examples.

Ans.4.(b) To demonstrate the Mapping of a Network Drive, the instructions below will show you how to connect to \\Galileo\ITD32. Please remember that all instructions below are for example only. You may well wish to connect to another drive but the instructions will act as a guide.

1. From the Start menu, choose Programs and choose Windows Explorer from the Submenu.
2. From the Tools menu, choose Map Network Drive... The window opposite will open.
3. Click the down arrow opposite Drive: and choose Z:



4. Click the down arrow opposite Folder:. Scroll through this list:-

(a) If the drive you wish to connect to is listed then click on it to connect to it. If it is not listed:

(b) Click Browse...the window "Browse for Folder" will open.

(i) Scroll through the list until you reach "Staff"

(ii) Expand the plus sign and scroll through this list until you reach "Galileo".

(iii) Expand this folder and choose ITD32

(iv) Click OK



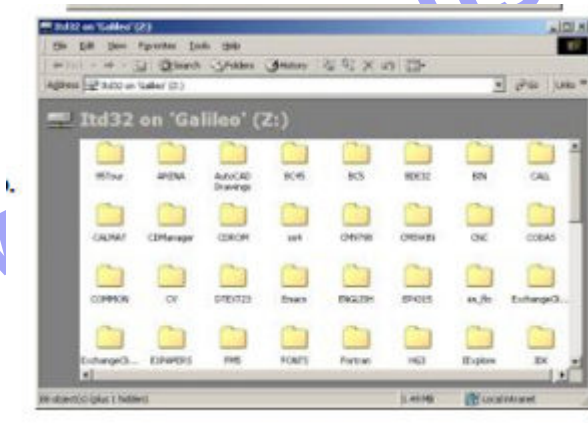


5. You will be returned to the Initial window – "Map Network Drive". The Folder location will now hold the drive you have connected to.

6. Click Finish.



7. Explorer will open showing you the contents of the drive you have connected to.



Q.5.(a)Mention the usage of following LINUX commands with an example of each :

a)tail

b)grep

c)chmod

d) sort

Ans.5.(a) Tail Command:-

## 1) About tail

tail outputs the last part, or "tail", of files.

## Description

tail prints the last 10 lines of each FILE to standard output. With more than one FILE, it precedes each set of output with a header giving the file name. If no FILE is specified, or if FILE is specified as a dash ("-"), tail reads from standard input.

## tail syntax

tail [OPTION]... [FILE]...

## tail examples

tail myfile.txt

Outputs the last 10 lines of the file myfile.txt.

tail myfile.txt -n 100

Outputs the last 100 lines of the file myfile.txt.

tail -f myfile.txt

Outputs the last 10 lines of myfile.txt, and monitors myfile.txt for updates; tail then continues to output any new lines that are added to myfile.txt.

tail -f access.log | grep 24.10.160.10

## 2) About grep

grep, which stands for "global regular expression print," processes text line by line and prints any lines which match a specified pattern.

### grep syntax

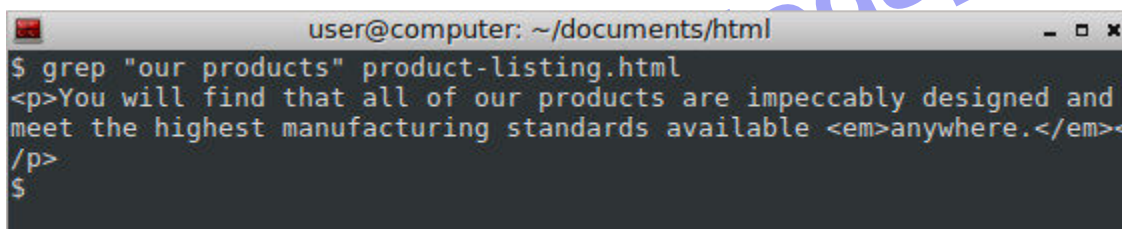
grep [OPTIONS] PATTERN [FILE...]

### Overview

Grep is a powerful tool for matching a [regular expression](#) against text in a file, multiple files, or a stream of input. It searches for the *PATTERN* of text that you specify on the command line, and outputs the results for you.

### Example Usage

Let's say want to quickly locate the phrase "our products" in HTML files on your machine. Let's start by searching a single file. Here, our *PATTERN* is "our products" and our *FILE* is product-listing.html.

A terminal window titled 'user@computer: ~/documents/html' showing a command and its output. The command is '\$ grep "our products" product-listing.html'. The output is '<p>You will find that all of our products are impeccably designed and meet the highest manufacturing standards available <em>anywhere.</em></p>'. The prompt '\$' is visible at the bottom.

```
user@computer: ~/documents/html
$ grep "our products" product-listing.html
<p>You will find that all of our products are impeccably designed and
meet the highest manufacturing standards available <em>anywhere.</em><
/p>
$
```

## 3) About chmod

chmod is used to change the [permissions](#) of [files](#) or [directories](#).

### Overview

On [Linux](#) and other [Unix-like operating systems](#), there is a set of rules for each file which defines who can access that file, and how they can access it. These rules are called file permissions or file *modes*. The command name chmod stands for "change mode", and it is used to define the way a file can be accessed.

Before continuing, you should read the section [What Are File Permissions, And How Do They Work?](#) in our documentation of the [umask](#) command. It contains a comprehensive description of how to define and express file permissions.

In general, chmod commands take the form:

For EX :- chmod options permissions filename

[Unix main page](#)

## 4)About sort

sort sorts the contents of a text file, line by line.

### Overview

lines starting with a number will appear before lines starting with a letter;

lines starting with a letter that appears earlier in the alphabet will appear before lines starting with a letter that appears later in the alphabet;

lines starting with a lowercase letter will appear before lines starting with the same letter in uppercase.

### sort syntax

sort sorts the contents of a text file, line by line.

sort is a simple and very useful command which will rearrange the lines in a text file so that they are sorted, numerically and alphabetically. By default, the rules for sorting are:

The rules for sorting can be changed according to the options you provide to the sort command; these are listed below.

sort [OPTION]... [FILE]...

sort [OPTION]... --files 0- from F Options

Q.5.(b) Compare the individual fields of the IPV4 header with the IPV6 header.

Ans.5.(b) If you are using Internet or almost any computer network you will likely be using IPv4 packets. IPv4 uses 32-bit source and destination address fields. We are actually running out of addresses but have not fear, the Internet Engineering Task Force (IETF) is here with IPv6. The IPv6 packet doesn't look much like its IPv4 cousin, except for the leading version field. The IPv6 address fields are 128-bits. The larger address space is one reason to migrate to IPv6 but there are many more differences that give IPv6 an advantage. For example, the header checksum field has been eliminated because transport reliability has gone up and its overhead was unnecessary.

	IPv4	IPv6
<b>Address</b>	32 bits (4 bytes) 1234:5678	128 bits (16 bytes) 1234:5678:9abc:def0
<b>Packet size</b>	576 bytes required, fragmentation optional	1280 bytes required without fragmentation
<b>Packet fragmentation</b>	Routers and sending hosts	Sending hosts only
<b>Packet header</b>	Does not identify packet flow for QoS handling Includes a checksum Includes options up to 40 bytes	Contains Flow Label field that specifies packet flow for QoS handling Does not include a checksum Extension headers used for optional data
<b>DNS records</b>	Address (A) records, maps host names Pointer (PTR) records, IN-ADDR.ARPA DNS domain	Address (AAAA) records, maps host names Pointer (PTR) records, IP6.ARPA DNS domain
<b>Address configuration</b>	Manual or via DHCP	Stateless address autoconfiguration (SLAAC) using Internet Control Message Protocol version 6 (ICMPv6) or DHCPv6
<b>IP to MAC resolution</b>	broadcast ARP	Multicast Neighbor Solicitation
<b>Local subnet group</b>	Internet Group Management	Multicast Listener Discovery (MLD)

The address space is the main difference between IPv4 (32-bit) and IPv6 (64-bit). The text representation has also been changed from a 2-digit partitioning for IPv4 to 4-digits for IPv6. An IPv4 example address is 12:34:56:78. An IPv6 example address is 1234:5678:9abc:def0:1234:5678:9abc:def0. The IPv6 representation also allows double colons (::) to represent a string of zero entries so 1234:0:9abc:0:0:0:0:def0 could be 1234:0:9abc::def0.

Larger data payloads can be shipped around the network by breaking the data among multiple packet fragments. This is typically done by the host but in IPv4 this can also be done by routers. IPv6 hosts need to determine the MTU for a path to a destination. This approach simplifies routers but adds complexity at the host end. This is normally not an issue and the IPv6 minimum MTU can always be used with any path.

Q.6.(a) Explain the Logical and Physical structure of active directory in Windows 2000.

Ans.6.(a) Active Directory can be considered to have both a logical and physical structure, and there is no correlation between the two. The logical parts of Active Directory include forests, trees, domains, OUs and global catalogs.

Each element of the logical structure of Active Directory is defined below:

**Domain** – a domain in Windows 2000 is very similar to a domain in Windows NT. It is still a logical group of users and computers that share the characteristics of centralized security and administration. A domain is still a boundary for security – this means that an administrator of a domain is an administrator for only that domain, and no others, by default. A domain is also a boundary for replication – all domain controllers that are part of the same domain must replicate with one another. Domains in the same forest automatically have trust relationships configured.

**Tree** – a tree is a collection of Active Directory domains that share a contiguous namespace. In this configuration, domains fall into a parent-child relationship, which the child domain takes on the name of the parent.

**Forest** – a forest is the largest unit in Active Directory and is a collection of trees that share a common Schema, the definition of objects that can be created. In a forest all trees are connected by transitive two-way trust relationships, thus allowing users in any tree access to resources in another for which they have been given appropriate permissions and rights. By default the first domain created in a forest is referred to as the root domain. Amongst other things, this is where the Schema is stored by default.

There are two types of active directory forest :-

- 1) Single Forest
- 2) Multiple forest

**Organizational Unit** – An organizational unit (OU) is a container object that helps to organize objects for the purpose of administration or group policy application. An OU exists within a domain and can only contain objects from that domain. OU can be nested, which allows for more flexibility in terms of administration. Different methods for designing OU structures exist including according to administration (most common), geography, or organizational structure. One popular use of OUs is to delegate administrative authority – this allows you to give a user a degree of administrative control over just the OU, and not the entire domain.

**Global Catalogs** – Global Catalogs are listings of every object that exists within an Active Directory forest. By default, a domain controller only contains information about objects in that domain. A Global Catalog server is a domain controller that contains information about every object (though not every attribute for each) stored in the entire forest.

The physical structure of Active Directory helps to manage the communication between servers with respect to the directory. The two physical elements of Active Directory are domain controllers and sites. Each is described below.

**Domain Controllers** – domain controllers are Windows 2000 Server-based systems that store the Active Directory database. Every Windows 2000 domain controller has a writable copy of the directory. This is different than in NT 4, where only the PDC had this capability. Domain controllers in the same domain contain replicas of the directory that must be synchronized periodically.

**Site** – a site is a concept that did not exist in an NT directory service structure. In Active Directory, sites are groups of IP subnets that are connected at high speed. Although the definition of 'high speed' is open, it is generally considered to be subnets that are connected at LAN speeds (say 10

Mb) or higher. The purpose of defining sites in Active Directory is to control network traffic relating to directory synchronization, as well as to help ensure that users connect to local resources. For example, domain controllers located in the same site replicate with one another on a 5-minute change notification interval similar to in NT 4.

Q.6.(b) Describe the concept of encrypting using EFS services.

Ans.6.(b)

## An Overview of the Encrypting File System

The Encrypting File System (EFS) is a component of the NTFS file system on Windows 2000, Windows XP Professional, and Windows Server 2003. (Windows XP Home doesn't include EFS.) EFS enables transparent encryption and decryption of files by using advanced, standard cryptographic algorithms. Any individual or program that doesn't possess the appropriate cryptographic key cannot read the encrypted data. Encrypted files can be protected even from those who gain physical possession of the computer that the files reside on. Even persons who are authorized to access the computer and its file system cannot view the data. While other defensive strategies should be used, and encryption isn't the correct countermeasure for every threat, encryption is a powerful addition to any defensive strategy. EFS is the built-in file encryption tool for Windows file systems.

However, every defensive weapon, if used incorrectly, carries the potential for harm. EFS must be understood, implemented appropriately, and managed effectively to ensure that your experience, the experience of those to whom you provide support, and the data you wish to protect aren't harmed. This document will

Provide an overview and pointers to resources on EFS.

Point to implementation strategies and best practices.

Name the dangers and counsel mitigation and prevention from harm.

What EFS Is

You can use EFS to encrypt files stored in the file system of Windows 2000, Windows XP Professional, and Windows Server 2003 computers. EFS isn't designed to protect data while it's transferred from one system to another. EFS uses symmetric (one key is used to encrypt the files) and asymmetric (two keys are used to protect the encryption key) cryptography. An excellent primer on [cryptography](#) is available in the Windows 2000 Resource Kit as is an introduction to [Certificate Services](#). Understanding both of these topics will assist you in understanding EFS.

The following are important basic facts about EFS:

EFS encryption doesn't occur at the application level but rather at the file-system level; therefore, the encryption and decryption process is transparent to the user and to the application. If a folder is marked for encryption, every file created in or moved to the folder will be encrypted. Applications don't have to understand EFS or manage EFS-encrypted files any differently than unencrypted files. If a user attempts to open a file and possesses the key to do so, the file opens without additional



effort on the user's part. If the user doesn't possess the key, they receive an "Access denied" error message.

File encryption uses a symmetric key, which is then itself encrypted with the public key of a public key encryption pair. The related private key must be available in order for the file to be decrypted. This key pair is bound to a user identity and made available to the user who has possession of the user ID and password. If the private key is damaged or missing, even the user that encrypted the file cannot decrypt it. If a recovery agent exists, then the file may be recoverable. If key archival has been implemented, then the key may be recovered, and the file decrypted. If not, the file may be lost. EFS is an excellent file encryption system—there is no "back door."

File encryption keys can be archived (e.g. exported to a floppy disk) and kept in a safe place to ensure recovery should keys become damaged.

EFS keys are protected by the user's password. Any user who can obtain the user ID and password can log on as that user and decrypt that user's files. Therefore, a strong password policy as well as strong user education must be a component of each organization's security practices to ensure the protection of EFS-encrypted files.

Q.7.(a) Assume you are server administrator of Linux lab. This lab is having two computers which are having some confidential data. We want to display a logon warning message, "unauthorized access to this system is punishable" if any user tries to log in these two computers with wrong user name or wrong password. Write the steps to create the above mentioned logon warning message.

Ans.7.(a) We can restrict the authentication of the User to save the confidential data from the Computers using CHMOD Command.

chmod - To change access permissions, change mode.

SYNOPSIS `chmod [Options]... Mode [,Mode]... file...` `chmod [Options]... Numeric_Mode file...` `chmod [Options]... --reference=RFile file...` DESCRIPTION

Numeric mode

EXAMPLES

```
$ chmod 400 sample.txt
$ chmod 040 sample.txt
$ chmod 004 sample.txt
$ chmod 200 sample.txt
$ chmod 020 sample.txt
$ chmod 002 sample.txt
$ chmod 100 sample.txt
$ chmod 010 sample.txt
$ chmod 001 sample.txt
$ chmod 444 sample.txt
$ chmod 777 sample.txt
```

Symbolic mode

EXAMPLES

```
$ chmod a-x sample.txt
$ chmod a+r sample.txt
```



```
$ chmod go+rw sample.txt
$ chmod u+x samplescript.sh
$ chmod =rw,g+s samplescript.sh
```

chmod changes the permissions of each given file according to mode, where mode describes the permissions to modify. Mode can be specified with octal numbers or with letters.

The format of a numeric mode is 'augo'

A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1. Any omitted digits are assumed to be leading zeros. The first digit selects the set user ID (4) and set group ID (2) and sticky (1) attributes. The second digit selects permissions for the user who owns the file: read (4), write (2), and execute (1); the third selects permissions for other users in the file's group, with the same values; and the fourth for other users not in the file's group, with the same values.

Read by owner only

Read by group only

Read by anyone

Write by owner only

Write by group only

Write by anyone

Execute by owner only

Execute by group only

Execute by anyone

Allow read permission to owner and group and anyone.

Allow everyone to read, write, and execute file.

The format of a symbolic mode is '[ugoa...][[+|=][rwxXstugo...]]...[,...]'. Multiple symbolic operations can be given, separated by commas. A combination of the letters 'ugoa' controls which users' access to the file will be changed: the user who owns it (u), other users in the file's group (g), other users not in the file's group (o), or all users (a). If none of these are given, the effect is as if 'a' were given, but bits that are set in the umask are not affected.

The operator '+' causes the permissions selected to be added to the existing permissions of each file; '-' causes them to be removed; and '=' causes them to be the only permissions that the file has.

The letters 'rwxXstugo' select the new permissions for the affected users: read (r), write (w), execute (or access for directories) (x), execute only if the file is a directory or already has execute permission

for some user (X), set user or group ID on execution (s), sticky (t), the permissions granted to the user who owns the file (u), the permissions granted to other users who are members of the file's group (g), and the permissions granted to users that are in neither of the two preceding categories (o).

Deny execute permission to everyone.

Allow read permission to everyone.

Make a file readable and writable by the group and others.

Make a shell script executable by the user/owner.

Allow everyone to read, write, and execute the file and turn on the set group-ID.

And after that print the message "unauthorized access to this system is punishable" using scripting .

Q.7.(b) Draw a diagram of SNMP architecture and show how it is used to manage network devices.

Ans.7.(b) SNMP Architecture:-

Simple Network Management Protocol (SNMP) is an application-layer protocol defined by the Internet Architecture Board (IAB) in RFC1157 for exchanging management information between network devices. It is a part of Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite.

SNMP is one of the widely accepted protocols to manage and monitor network elements. Most of the professional-grade network elements come with bundled SNMP agent. These agents have to be enabled and configured to communicate with the network management system (NMS).

SNMP basic components and their functionalities

SNMP Manager:

A manager or management system is a separate entity that is responsible to communicate with the SNMP agent implemented network devices. This is typically a computer that is used to run one or more network management systems.

SNMP Manager's key functions

Queries agents

Gets responses from agents

Sets variables in agents

Acknowledges asynchronous events from agents

Managed Devices:

A managed device or the network element is a part of the network that requires some form of monitoring and management e.g. routers, switches, servers, workstations, printers, UPSs, etc...

### SNMP Agent:

The agent is a program that is packaged within the network element. Enabling the agent allows it to collect the management information database from the device locally and makes it available to the SNMP manager, when it is queried for. These agents could be standard (e.g. Net-SNMP) or specific to a vendor (e.g. HP insight agent)

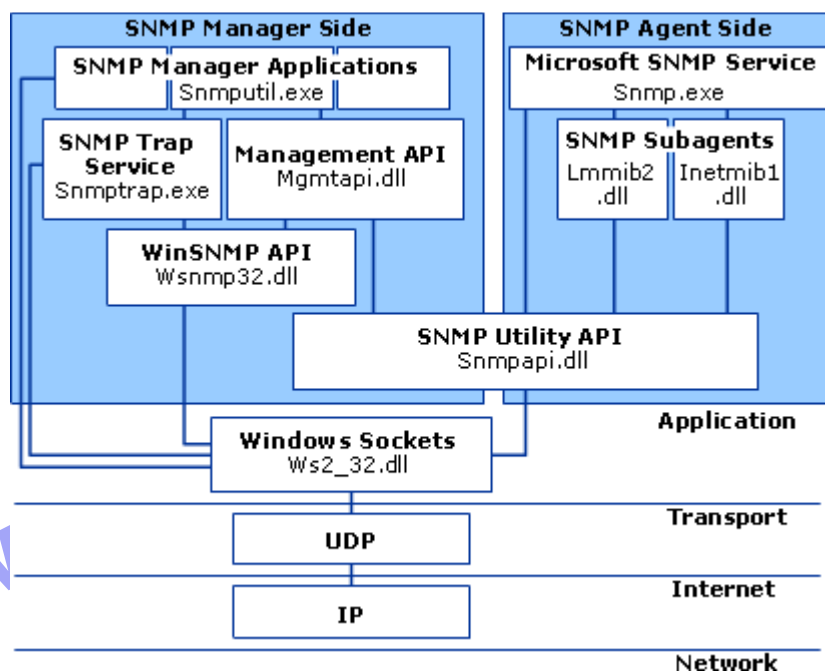
### SNMP agent's key functions

Collects management information about its local environment

Stores and retrieves management information as defined in the MIB.

Signals an event to the manager.

Acts as a proxy for some non-SNMP manageable network node.



Q.8.(a) How does the Remote Access Mechanism in Windows 2000 work and how can it be configured ?

Ans.8.(a) The term “remote access server” can refer to a server that performs a range of remote access services, instead of just providing the ability for clients to dial into the company LAN. Both Windows 2000 Professional and Windows 2000 Server can act as remote access servers, albeit with different restrictions on each platform.

On the Windows 2000 Professional side, you can configure a workstation to allow incoming connections through dial-up (one at a time), giving the remote caller the ability to use resources stored on the local computer or on the LAN, depending on how RRAS is configured. Under Windows 2000 Server, RRAS can support multiple concurrent remote access clients for those same purposes, essentially limited only by the number of available incoming connections. For example, if you have a modem pool of 48 modems, Windows 2000 Server will support all of those connections concurrently.

A RAS connection that connects the client to the dial-up server is called a point-to-point remote access connection. A RAS connection that connects the client to the LAN is called a point-to-LAN remote access connection. Regardless of the type, the remote clients can access resources on the server or LAN as if their computers were connected locally to the server or LAN. For example, clients can open and save files and use printers, just as they can locally.

Remote access is not the same thing as remote control. With a remote control application such as Symantec's pcAnywhere, the client uses the remote control application to log onto and run applications on a remote computer. The applications run on the remote computer rather than on the client's local computer. In effect, the remote control application gives the client a long-distance keyboard, mouse, and display for the remote computer.

Remote access makes the client's local computer a part of the remote network. Applications run on the client's local computer, not on the remote computer (except when the client executes a network-enabled application). Remote control applications can't exist without remote access—the client either dials into the remote computer directly or dials into the LAN. So, if you need to use a remote control application to manage a remote server, for example, you'll need a remote access connection to the server or to the server's LAN before the remote control application can do anything. Depending on the remote control application, that connection might take the form of a public Internet connection, using the remote server's and client's existing connections to the Internet as the means of communication.

### Setting up the hardware

Whether you're configuring a Windows 2000 Professional computer to enable single connections or a Server computer to handle a modem pool, your first step in setting up a remote access server is to configure the hardware for the incoming connections. These connections might come in through one or more modems connected to the computer's communications ports, through a multiport communications card handling multiple modems, a modem pool/communications server connected to the LAN, or even a network interface.

While you can certainly grow the server's capabilities later on, you need to determine your clients' current needs and plan for that growth. Choosing the right communications hardware is a big part of that process. If the bandwidth needs aren't critical, modems and Public Switched Telephone Network (PSTN) lines (standard voice lines, or Plain Old Telephone Service—POTS) are an easy and relatively inexpensive solution. If you're installing multiple lines, choose one number as the primary dial-up number and have your communications provider configure the lines in a hunt group. If one line is busy in the hunt group, the incoming call rolls to the next available line. There are

several options for hunt groups that can address such problems as a ring-no-answer due to a hung modem. Check with your provider for details to decide what best fits your needs.

The Routing and Remote Access service for Windows 2000 Server continues the evolution of multiprotocol routing and remote access services for the Microsoft Windows platform. New features of the Routing and Remote Access service for Windows 2000 include:

Internet Group Management Protocol (IGMP) and support for multicast boundaries.

Network address translation with addressing and name resolution components that simplify the connection of a small office/home office (SOHO) network to the Internet.

Integrated AppleTalk routing.

Layer Two Tunneling Protocol (L2TP) over IP Security (IPSec) support for router-to-router VPN connections.

Improved administration and management tools. The graphical user interface program is the Routing and Remote Access administrative utility, a Microsoft Management Console (MMC) snap-in. The command-line utility is Netsh.

All of the combined features of the Windows 2000 Routing and Remote Access service make a Windows 2000 Server-based computer function as the following:

Multiprotocol router

A Routing and Remote Access service computer can route IP, IPX, and AppleTalk simultaneously. All routable protocols and routing protocols are configured from the same administrative utility.

Demand-dial router

A Routing and Remote Access service computer can route IP and IPX over on-demand or persistent WAN links, such as analog phone lines or ISDN, or over VPN connections using either PPTP or L2TP over IPSec.

Remote access server

A Routing and Remote Access service computer can act as a remote access server providing remote access connectivity to dial-up or VPN remote access clients using IP, IPX, AppleTalk, or NetBEUI.

Q.8.(b) Explain the following with reference to WINDOWS 2000:

(i) File Replication service

(ii) FAT 16 and FAT 32

Ans.8.(b) i) File Replication Service:-

File Replication Service (FRS) is a feature in Microsoft Windows Server which is a successor to the LAN Manager Replication service of Windows NT Server. It is used for the replication of the system policies and script by the Windows Server. This data is stored in the SYSVOL, or the system volume, of the server. It is stored in the controllers of the domain, and can be accessed by the client servers of the network. Distributed File System Replication Service is now quickly replacing File Replication Service.

FRS is a service which allows the sharing of Group Policies and logon scripts to the domain controllers, from where they may be accessed by the users through the client servers. The executable file running the service is NTFRS.exe. This service can also be used to replicate files and synchronize the data of its domain controllers using a DFS. It is also able to keep data on multiple servers at once.

The synchronization process is quick and complete. As it initiates very important scripts and processes which are required for logon, the services have to be quick, efficient and dependable. This service fits all the requirements because it backs up all the data on different servers while replicating them. The sync service is very fast and any changes in the policies are instantaneously changed in the client's data.

## ii) FAT 16 and FAT 32:- Differences Between FAT 16 and FAT 32

Drive Size	Default FAT16 Cluster Size	Default FAT32 Cluster Size
260 MB–511 MB	8 KB	Not supported
512 MB–1,023 MB	16 KB	4 KB
1,024 MB–2 GB	32 KB	4 KB
2 GB–8 GB	Not supported	4 KB
8 GB–16 GB	Not supported	8 KB
16 GB–32 GB	Not supported	16 KB
> 32 GB	Not supported	32 KB

There are additional differences between FAT32 and FAT16:

FAT32 allows finer allocation granularity (approximately 4 million allocation units per volume).

FAT32 allows the root directory to grow (FAT16 holds a maximum of 512 entries, and the limit can be even lower due to the use of long file names in the root folder).

Advantages of FAT16 are:

MS-DOS, Windows 95, Windows 98, Windows NT, Windows 2000, and some UNIX operating systems can use it.

There are many tools available to address problems and recover data.

If you have a startup failure, you can start the computer with an MS-DOS bootable floppy disk.

It is efficient, both in speed and storage, on volumes smaller than 256 MB.

## **Disadvantages of FAT16**

Disadvantages of FAT16 are:

The root folder can manage a maximum of 512 entries. The use of long file names can significantly reduce the number of available entries.

FAT16 is limited to 65,536 clusters, but because certain clusters are reserved, it has a practical limit of 65,524. Each cluster is fixed in size relative to the logical drive. If both the maximum number of clusters and their maximum size (32 KB) are reached, the largest drive is limited to 4 GB on Windows 2000. To maintain compatibility with MS-DOS, Windows 95, and Windows 98, a FAT16 volume should not be larger than 2 GB.

The boot sector is not backed up.

There is no built-in file system security or file compression with FAT16.

FAT16 can waste file storage space in larger drives as the size of the cluster increases. The space allocated for storing a file is based on the size of the cluster allocation granularity, not the file size. A 10-KB file stored in a 32-KB cluster wastes 22 KB of disk space.

## **Advantages of FAT32**

FAT32 allocates disk space much more efficiently than previous versions of FAT. Depending on the size of your files, there is a potential for tens and even hundreds of megabytes more free disk space on larger hard disk drives. In addition, FAT32 provides the following enhancements:

The root folder on a FAT32 drive is now an ordinary cluster chain, so it can be located anywhere on the volume. For this reason, FAT32 does not restrict the number of entries in the root folder.

It uses space more efficiently than FAT16. FAT32 uses smaller clusters (4 KB for drives up to 8 GB), resulting in 10 to 15 percent more efficient use of disk space relative to large FAT16 drives. FAT32 also reduces the resources necessary for the computer to operate.

FAT32 is more robust than FAT16. FAT32 has the ability to relocate the root directory and use the backup copy of the FAT instead of the default copy. In addition, the boot record on FAT32 drives has been expanded to include a backup of critical data structures. This means that FAT32 volumes are less susceptible to a single point of failure than FAT16 volumes.

## Disadvantages of FAT32

Disadvantages of FAT32 include: The largest FAT32 volume Windows 2000 can format is limited in size to 32 GB. FAT32 volumes are not accessible from any other operating systems other than Windows 95 OSR2 and Windows 98. The boot sector is not backed up. There is no built-in file system security or compression with FAT32.

[www.ignou.site.blogspot.com](http://www.ignou.site.blogspot.com)



[www.ignou.site.blogspot.com](http://www.ignou.site.blogspot.com)