Q#1 **what are Interrupts & Identify I/O Device?**

Interrupt are an important part of computer Architecture. Each computer design has its own interrupts mechanism, But several function are common.

**OR**

A **hardware interrupt** is an electronic alerting signal sent to the processor from an external device, either a part of the computer itself such as a disk controller or an external peripheral. For example, pressing a key on the keyboard or moving the mouse triggers hardware interrupts that cause the processor to read the keystroke or mouse position

**I/O Devices:**

In computing, **input/output** or **I/O** is the communication between information processing system (such as a computer) and the outside world, possibly a human or another information processing system? Inputs are the signals or data received by the system, and outputs are the signals or data sent from it.

**Input devices**:

* keyboards,
* numeric keypads,
* pointing devices (including mouse, touch pad and tracker ball),
* remote controls,
* joysticks,

**Output devices**:

* monitors (CRT, TFT),
* printers (laser, ink jet and dot matrix),
* plotters,
* speakers,

**Q#2 Describe Structure of I/O system?**

I/O Structure

♦After I/O starts, control returns to user program only upon I/O completion.

–wait instruction idles the CPU until the next interrupt

–wait loop (contention for memory access).

–At most one I/O request is outstanding at a time, no simultaneous I/O processing.

♦After I/O starts, control returns to user program without waiting for I/O completion.

–*System call*–request to the operating system to allow user to wait for I/O completion.

–*Device-status table* contains entry for each I/O device indicating its type, address, and state.

–Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt.

**Q#3 Name the different Operating System describes one specific characteristics of each that we can use?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operating System** | Features | Description | Benefits |
| Windows Xp | **.Built on the Windows Engine**  **Enhanced Device Driver Verifier**  **Dramatically Reduced Reboot Scenarios**  **Improved Code Protection** | Windows XP Professional is built on the proven code base of Windows NT® and Windows 2000, which features a 32-bit computing architecture and a fully protected memory model. | Windows XP Professional will provide a dependable computing experience for all business users. |
| Linux operating system | • Multitasking: Several programs can run at the same time.  •Multiuser: Several users can logon to the same machine at the same time There is no need to have separate user licenses.  • Multiplatform: Linux runs on many different CPUs that mean it supports multiprocessor machine. | Linux was originally developed as a free operating system for Intel x86-based personal computers. It is a leading operating system on servers and other big iron systems such as mainframe computers and supercomputers | * Low cost * Stability * Performance * Network friendliness |
| Windows 8 | **Windows 8** includes new features, including native USB 3.0 and full 4K Advanced Format support, Microsoft account integration, Windows Store, the ability to boot from USB Flash drives with Windows To Go, and easier system restore options, among others | **Windows 8** is a version of Microsoft Windows for use on personal computers, including home and business desktops, laptops, tablets, and home theater PCs. | **1. Faster startup**  **2. A whole new world of apps.**  **3. Sky Drive integration**  **4. Better Security, Less-intrusive updates.**  **5. First-class touch input, but still fine with keyboard and mouse.** |

**Q#4Which operating system is best in your opinion?**

**Windows 8:-**

Windows 8 is a version of Microsoft Windows (an operating system developed by Microsoft) for use on personal computers, including home and business desktops, laptops, tablets, and home theater PCs. Development of Windows 8 started before the release of its predecessor, Windows 7, in 2009. It was announced at CES 2011, and followed by the release of three pre-release versions from September 2011 to May 2012. The operating system was released to manufacturing on August 1, 2012, and was released for general availability on October 26, 2012.Windows 8 uses version 6.2 of the Windows NT kernel.

**Benefits of Windows 8:-**

**1. Touch screen interface**

An obvious difference between Windows 8 and its predecessors is its completely revamped interface. The Modern UI (formerly known as “Metro”) is designed first and foremost with touch input in mind.

**2. Networking**

One of the most common headaches for Windows users—particularly mobile users trying to work from customer sites or remote locations—is finding and connecting to a network. Microsoft has made improvements in Windows 8 that make accomplishing this task easier and more intuitive.

**3. Flexible hardware options**

Since their inception, PCs and laptops have maintained a fairly consistent approach to form and function. Sure, they’ve gotten smaller over the years, but a desktop remained a desktop, and a laptop a laptop, more or less—until now.

**4. Faster boot time**

Though the usual waiting period is only a matter of seconds, it can feel like an eternity as a computer wakes up from a complete shutdown and finally reaches the Windows login screen—especially if you're at a meeting, where every second counts.

**5. Dual-monitor support**

It’s not exactly mainstream in most fields of business, but using multiple monitors can greatly improve productivity, and Windows 8 comes with a number of enhancements to simplify managing and using such setups.

**6. Better security**

Windows 8 introduces some security tricks to help protect data and let IT managers sleep at night. First, Microsoft takes advantage of the Secure Boot feature of UEFI (Unified Extensible Firmware Interface). Secure Boot allows only software signed by authorized certificates to boot up, which prevents BIOS- or kernel-level malware from sneaking in.

**7. Storage Spaces**

Hard drives keep getting larger and cheaper, but newer hardware such as Ultra books and tablets tend to rely on smaller-capacity solid-state drives for storage. Windows 8's Storage Spaces feature lets you expand your storage without replacing your drive, and without having to add new drive letters and then try to manage which applications or data get stored on which drive.

**8. Sky Drive integration**

As great as Storage Spaces can be, it works only if the various drives in the pool are connected to the Windows 8 PC. When you’re on the go, using the cloud to expand your storage options makes more sense. Microsoft has woven access to its cloud storage service, Sky Drive, throughout Windows 8.

**9. New Task Manager**

The Task Manager has always been a powerful but underused tool in Windows. In Windows 8, though, it receives a complete makeover that makes it both easier to use, and more valuable than ever.

**10. Windows to Go**

You’ll have to use Windows 8 Enterprise in order to take advantage of Windows to Go, but for many businesses, that version of Windows is worth its higher price. Windows to Go lets you store an entire Windows 8 environment on a bootable USB thumb drive or other removable media.

**Q#5 Explain the design principle of Unix/Linux?**

Design Principles

1. Linux is a multi-user, multitasking system with a full set of UNIX-compatible tools.

2. Its file system adheres to traditional UNIX semantics, and it fully implements the standard UNIX networking model.

3. Main design goals are speed, efficiency, and standardization.

4. Linux is designed to be compliant with the relevant POSIX documents; at least two. Linux distributions have achieved official POSIX certification.

5. Linux is composed of three main bodies of code; the most important distinction between the kernel and all other components.

6. The kernel is responsible for maintaining the important abstractions of the operating system.

   1. Kernel code executes in *kernel mode* with full access to all the physical resources of the computer.

  2. All kernel code and data structures are kept in the same single address space.

The system libraries define a standard set of functions through which applications interact with the kernel, and which implement much of the operating-system functionality that does not need the full privileges of kernel code.

The system utilities perform individual specialized management tasks.

**Q#6 Give brief history of Evolution of the Unix?**

**Introduction**

During the past few years, the Unix operating system has come into wide use, so wide that its very name

has become a trademark of Bell Laboratories. It’s important characteristics have become known to many

people. It has suffered much rewriting and tinkering since the first publication describing it in 1974 [1],

but few fundamental changes. However, Unix was born in 1969 not 1974, and the account of its

development makes a little-known and perhaps instructive story. This paper presents a technical and

social history of the evolution of the system.

**Overview**

Originally, UNIX was meant to be a programmer's workbench to be used for developing software to be run on multiple platforms more than to be used to run application software. The system grew larger as the operating system started spreading in the academic circle, as users added their own tools to the system and shared them with colleagues

**Unix file system**

Structurally, the file system of PDP-7 Unix was nearly identical to todays. It had

1) An I-list: a linear array of *I-nodes* each describing a file. An I-node contained less than it does

now, but the essential information was the same: the protection mode of the file, its type and size,

and the list of physical blocks holding the contents.

2) Directories: a special kind of file containing a sequence of names and the associated I-number.

3) Special files describing devices. The device specification was not contained explicitly in the

I-node, but was instead encoded in the number: specific I-numbers corresponded to specific files.

**Process control**

By ‘process control,’ I mean the mechanisms by which processes are created and used; today the system

calls *fork*, *exec*, *wait*, and *exit* implement these mechanisms. Unlike the file system, which existed in?

nearly its present form from the earliest days, the process control scheme underwent considerable

mutation after PDP-7 Unix was already in use.

Today, the way in which commands are executed by the shell can be summarized as follows:

1) The shell reads a command line from the terminal.

2) It creates a child process by *fork.*

3) The child process uses *exec* to call in the command from a file.

4) Meanwhile, the parent shell uses *wait* to wait for the child (command) process to terminate by

calling *exit.*

5) The parent shell goes back to step 1).

**High-level languages**

Every program for the original PDP-7 Unix system was written in assembly language, and bare

Assembly language it was for

example, there were no macros. Moreover, there was no loader or

link-editor, so every program had to be complete in itself. The first interesting language to appear was a

Version of McClure’s TMG [11] that was implemented by McElroy. Soon after TMG became available,

Thompson decided that we could not pretend to offer a real computing service without Fortran, so he sat

down to write a Fortran in TMG. As

**Q#7 Identify the hardware requirements for Linux/Unix?**

|  |  |
| --- | --- |
| **Component** | **Minimum Requirement** |
| Computer  **Processor**  **Memory**  Disk space: Interchange Server, databases and supporting software | p Series® 610 6E1 class or equivalent  375 MHz IBM POWER3™-II  512 MB main memory  **40 gb** |

**Q#8 Write the steps of adding and maintaining new user Linux operating System and installation of Unix?**

# Creating a User Account

There are two ways to create new and/or additional user accounts: using the graphical **User Manager** application or from a shell prompt.

To create a user account graphically using the **User Manager**:

1. Select **Applications** (the main menu on the panel) => **System Settings** => **Users & Groups** from the panel. You can also start the **User Manager** by typing red hat-configuration-users at a shell prompt.
2. If you are not logged in as root, you will be prompted for your root password.
3. A window will appear. Click **Add User**.
4. In the **Create New User** dialog box, enter a username (this can be an abbreviation or nickname), the full name of the user for whom this account is being created, and a password (which you will enter a second time for verification). The name of this user's home directory and the name of the login shell should appear by default. For most users, you can accept the defaults for the other configuration options.
5. Click **OK**. The new user will appear in the user list, signaling that the user account creation is complete.

To create a user account from a **shell prompt**:

1. Open a shell prompt.
2. If you are not logged in as root, type the command su - and enter the root password.
3. Type useradd followed by a space and the username for the new account you are creating at the command line (for example, useradd *jsmith*). Press [Enter]. Often, usernames are variations on the user's name, such as jsmith for John Smith. User account names can be anything from the user's name, initials, or birthplace to something more creative.
4. Type passwd followed by a space and the username again (for example, passwd *jsmith*).
5. At the New password: prompt enter a password for the new user and press [Enter].

At the Retype new password: prompt, enter the same password to confirm your selection.

**Maintaining User Accounts**

**Maintaining** attributes of user accounts and groups is usually a simple matter of editing /etc/passwd and /etc/group. Many systems provide commands such as usermod and groupmod to do just this; it's often easier to edit the files by hand.

To change a user's password, use the passwd command, which will prompt for a password, encrypt it, and store the encrypted password in the /etc/passwd file.

If you need to change the user ID of an existing account, you can do this by editing the uid field of /etc/passwd directly. However, you should also chown the files owned by the user to that of the new uid. For example:

chown -R aclark /home/aclark

**Q#9 Why is necessary to monitor the performance of UNIX System?**

**T**he performance analyst to some of the free tools available to monitor and manage performance on UNIX systems, and to provide a guideline on how to diagnose and fix performance problems in UNIX environment.

UNIX has following major resource types that need to be monitored and tuned:

* **CPU**
* **Memory**
* **Disk space**
* **Communications lines**
* **I/O Time**
* **Network Time**
* **Applications programs**

**Performance Components:**

|  |  |
| --- | --- |
|  |  |

1. **User state CPU**: The actual amount of time the CPU spends running the users program in the user state. It includes time spent executing library calls, but does not include time spent in the kernel on its behalf.
2. **System state CPU:** This is the amount of time the CPU spends in the system state on behalf of this program. All I/O routines require kernel services. The programmer can affect this value by the use of blocking for I/O transfers.
3. **I/O Time and Network Time:** These are the amount of time spent moving data and servicing I/O requests
4. **Virtual Memory Performance:** This includes context switching and swapping.
5. **Application Program:** Time spent running other programs - when the system is not servicing this application because another application currently has the CPU.

**Q#10Which tools are used for performance monitoring?**

**Performance Tools:**

UNIX provides following important tools to measure and fine tune UNIX system performance:

|  |  |
| --- | --- |
| **nice/renice :** | Run a program with modified scheduling priority |
| **netstat :** | Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships |
| **Time:** | Time a simple command or give resource usage |
| **Uptime:** | System Load Average |
| **Ps:** | Report a snapshot of the current processes. |
| **Vmstat:** | Report virtual memory statistics |
| **Gprof:** | Display call graph profile data |
| **Prof:** | Process Profiling |
| **Top:** | Display system tasks |

**Q#11How many types of commands are there in Linux? Name the category and briefly explain them.**

There are two types of commands in Linux

1. Linux commands
2. Shell Commands

**Linux commands:-**

Linux commands are the same whatever type of shell you are using, but differ between Linux distributions (Red hat, Mandrake, SUSE, Debian etc). We have compiled a list of Linux commands specific to Red hat 9.0, and are in the process of writing easy to understand explanations for each command.

**Shell commands:-**

Shell commands vary from shell to shell (bash shell, borne shell, C shell, korn shell etc), but remain the same across Linux distributions. As with the Linux commands, we have compiled a list of bash shell commands, and are working our way through writing the basic explanations for those commands.

**Q#12 Write syntax, usage and description of printing and scheduling commands of UNIX?**

**Print Commands:-**

**Usage of Printing Command**:

Following tools use for managing printing   
  
lpq: Checks the status of print jobs.   
lprm: Removes a print job from a spool.   
lpstat: Reads the status of a spool.

**Syntax:-**

*lpr [ -P destination ] [ -# number ] [ -C class] [ -J job ] [ -T title ] [ -i [ indent ] ] [-1 | -2 | -3 | -4 font ] [ -w cols ] [ -m ] [ -h ] [ -s ] [-filter\_option ] [ file ... ]*

**DESCRIPTION**

**lpstat** displays status information about the current classes, jobs, and printers. When run with no arguments, **lpstat** will list jobs queued by the user. Other options include:

-E

Forces encryption when connecting to the server.

**Scheduling Commands:-**

**Usage of Scheduling Commands:**

A Linux system can have a lot to suffer from, but it usually suffers only during office hours. Whether in an office environment, a server room or at home, most Linux systems are just idling away during the morning, the evening, the nights and weekends

**Syntax:-**

.at

at -f shell\_script now + 1 hour

**Example:-**

at -f shell\_script now + 1 hour

**Q#13 what is meant by Backup? Why it is necessary to keep the backup update?**

# Backup

In information technology, a **backup**, or the process of **backing up**, refers to the copying and archiving of computer data so it may be used to *restore* the original after a data loss event. The verb form is **to back up** in two words, whereas the noun is *back up*.

Backups have two distinct purposes. The primary purpose is to recover data after its loss, be it by data deletion or corruption. Data loss can be a common experience of computer users

The system state data can be backed up in any order. Restoration of the system state replaces boot files first and commits the system hive of the registry as a final step in the process.

System state backup and restore operations include all system state data: you cannot choose to backup or restore individual components due to dependencies among the system state components. However, you can restore system state data to an alternate location in which only the registry files, Sysvol directory files, and system boot files are restored. The Active Directory database, Certificate Services database, and Component Services Class Registration database are not restored to the alternate location.

Although you cannot change which components of the system state are backed up, you can back up all protected system files with the system state data by setting advanced backup options.

**Q#14Write the steps of back up and restoration of UNIX/Linux system?**

**Backing up System**

1. Perform a full database backup.
2. Back up the home directories of the operating system users: <sapsid>adm
3. Database-specific operating system user(s)

Back up the following file systems of your system:

/usr/sap/<SAPSID>

/usr/sap/trans

/<sapmnt>/<SAPSID>

**Proceed as follows to perform the backup:**

* Log on as user root
* Manually create a compressed tar archive that contains all installed files:

**Saving to tape:**

tar —cf — <file\_system> | compress —c> <tape\_device>

**Saving to the file system:**

tar —cf — <file\_system> | compress —c> ARCHIVENAME.tar.Z.

**4. Back up the operating system using operating system tools.**

This saves the structure of the system and all configuration files, such as file system size, configuration of the logical volume manager, and database configuration data.

**Restoring the Backup**

If required, you can restore the data that you previously backed up.

1. Log on as user root.
2. Go to the location in your file system where you want to restore the backup image.
3. Proceed as follows to restore the backup:

* Execute the following command to restore the data from tape:

cat <tape\_device> | compress —cd | tar —xf —

* Execute the following command to restore the data from the file system:

cat ARCHIVENAME.tar.Z | compress —cd | tar —xf —

**4**. Restore your database backup.

**Q#15 Write short note on any of the following:**

1. **Bourne Shell**
2. **Kernel**
3. **FTP protocol**
4. **VI Editor**

**Bourne Shell:-**

The **Bourne shell** (**sh**) is a shell (computing), or command-line interpreter, for computer operating systems.

The Bourne shell was the default UNIX shell of Unix Version 7. Most Unix-like systems continue to have /bin/sh—which will be the Bourne shell, or a symbolic link or hard link to a compatible shell even when other shells are used by most users.

Developed by Stephen Bourne at Bell Labs, it was a replacement for the Thompson shell, whose executable file had the same name—sh. It was released in 1977 in the Version 7 Unix release distributed to colleges and universities. Although it is used as an interactive command interpreter, it was always intended as a scripting language and contains most of the features that are commonly considered to produce structured programs.

1. **Kernel**

In computing, the **kernel** is the main component of most computer operating systems; it is a bridge between applications and the actual data processing done at the hardware level. The kernel's responsibilities include managing the system's resources (the communication between hardware and software components).Usually, as a basic component of an operating system, a kernel can provide the lowest-level abstraction layer for the resources (especially processors and I/O devices) that application software must control to perform its function. It typically makes these facilities available to application processes through inter-process communication mechanisms and system calls.

**FTP protocol**

**File Transfer Protocol** (**FTP**) is a standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet.

FTP is built on client-server architecture and uses separate control and data connections between the client and the server.FTP users may authenticate themselves using a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that hides (encrypts) the username and password, and encrypts the content, FTP is often secured with SSL/TLS ("FTPS"). SSH File Transfer Protocol ("SFTP") is sometimes also used instead, but is technologically different.

FTP is the easiest way to transfer files between computers via the internet, and utilizes TCP, transmission control protocol, and IP, internet protocol, systems to perform uploading and downloading tasks.

**VI Editor:-**

The vi editor (short for visual editor) is a screen editor which is available on almost all Unix systems.vi has no menus but instead uses combinations of keystrokes in order to accomplish commands.

VI is available by default on every Unix machine Because of its universal availability.

Different vi clones of course have different ways of starting the program (invocation). Usually, however, command-line versions of vi share a common basic set of command line options. These following command line options and flags are typically available. In addition, vi can be started under different names. Depending on the name used to start vi, it may either behave slightly differently or load a different vi clone.

The common command line options and flags are

-

or

-s

**Q#16 Define System Calls?**

**System Calls:-**

A system call is a mechanism that is used by the application program to request a service from the operating system. They use a machine-code instruction that causes the processor to change mode. An example would be from supervisor mode to protected mode. This is where the operating system performs actions like accessing hardware devices or the memory management unit. Generally the operating system provides a library that sits between the operating system and normal programs. Usually it is a C library such as Glibc or Windows API. The library handles the low-level details of passing information to the kernel and switching to supervisor mode. System calls include close, open, read, wait and write.

There are 5 different categories of system calls:

* 1. Process control
  2. file manipulation
  3. device manipulation
  4. information maintenance
  5. Communication.

**Process Control**

A running program needs to be able to stop execution either normally or abnormally. When execution is stopped abnormally, often a dump of memory is taken and can be examined with a debugger.

**File Management**

Some common system calls are create, delete, read, write, reposition, or close. Also, there is a need to determine the file attributes – get and set file attribute. Many times the OS provides an API to make these system calls.

**Device Management**

Process usually requires several resources to execute, if these resources are available, they will be granted and control returned to the user process. These resources are also thought of as devices. Some are physical, such as a video card, and others are abstract, such as a file.

User programs request the device, and when finished they release the device. Similar to files, we can read, write, and reposition the device.

**Information Management**

Some system calls exist purely for transferring information between the user program and the operating system. An example of this is time, or date.

The OS also keeps information about all its processes and provides system calls to report this information.

**Communication**

There are two models of inter process communication, the message-passing model and the shared memory model.

* Message-passing uses a common mailbox to pass messages between processes.
* Shared memory use certain system calls to create and gain access to create and gain access to regions of memory owned by other processes. The two processes exchange information by reading and writing in the shared data.

**Q#17 Explain how operating System manages the resources of a computer? (2005, 2008)**

An **operating system** (**OS**) is a collection of software that manages computer hardware resources and provides common services for computer programs. The operating system is a vital component of the system software in a computer system. Application programs usually require an operating system to function.

**an operating system does two things:**

1. It manages the hardware and software resources of the system. In a desktop computer, these resources include such things as the processor, memory, disk space and more

2. It provides a stable, consistent way for applications to deal with the hardware without having to know all the details of the hardware.

**The following Elements are generally included in this set of software:**

**Kernel:-**

The **kernel** is a bridge between applications and the actual data processing done at the hardware level.

A kernel is like the heart of an OS. It allocates Hardware resources to the Programs. For Eg: if your program asks needs a printer to print a document it requests a printer to the OS. That request is processed by kernel and the printer is allocated to the Program.

**Shell:-**

Shells are actually special applications which use the kernel API in just the same way as it is used by other application programs. A shell manages the user–system interaction by prompting user(s) for input, interpreting their input, and then handling an output from the operating system.

**File System:-**

File system in OS provide the way to create and access the files.

How the memory is allocated to files and how the addresses are saved regarding to each file.

All this is about the address translation and calculation for files.

The operating system acts as an interface between the hardware and the programs requesting I/O. It is the most fundamental of all system software programs.

Responsibilities of the OS include:

* Hiding the complexities of hardware from the user
* Managing between the hardware's resources which include the processors, memory, data storage and I/O devices
* Handling "interrupts" generated by the I/O controllers
* Sharing of I/O between many programs using the CPU

The most well known Operating Systems include:

* **System Software** - programs that manage the operation of a computer
* **Application Software** - programs that help the user perform a particular task

**Q#18 differentiates? (2008)**

**Fat and NTFS**

**FAT**File Allocation Table is a primary computer file system for various operating systems, mostly DOS, including DR-DOS, OpenDOS, freeDOS, MS-DOS, Microsoft Windows (up to and including Windows Me). FAT is also used for removable flash drives and memory cards.  
  
File system is a method for storing and organizing computer files and the data they contain to make it easy to find and access them. The chart below shows in what FAT system a flash drive or memory card should be formatted.

**NTFS**  
NTFS is the standard file system of Windows NT, including its later versions Windows 2000, Windows XP, Windows Vista, Windows Server 2003, and Windows Server 2008. NTFS is intended for use in Windows system drives (Hard Disk Drives and Solid State Drives).  
  
NTFS has several improvements over FAT such as improved support for metadata and the use of advanced data structures for reliability, and disk space utilization, plus additional extensions such as security access control lists and file system journaling.

**Q#19 Describe the function of an operating system?**

**1. Manages and Interacts with Computer Hardware**

* Availability
  + Concerned with protecting the system against interruption
* Confidentiality
  + Assuring that users cannot read data for which access is unauthorized

**2. Provides and Manages System Security**

* Fairness
  + Give equal and fair access to resources
* Differential responsiveness
  + Discriminate among different classes of jobs
* Efficiency

Maximize throughput, minimize response time, and accommodate as many uses as possible

**3. Provides the System Interface**

an **interface** is a tool and concept that refers to a point of interaction between components, and is applicable at the level of both hardware and software. This allows a component, whether a piece of hardware such as a graphics card or a piece of software such as an Internet browser, to function independently while using interfaces to communicate with other components via an input/output system and an associated protocol

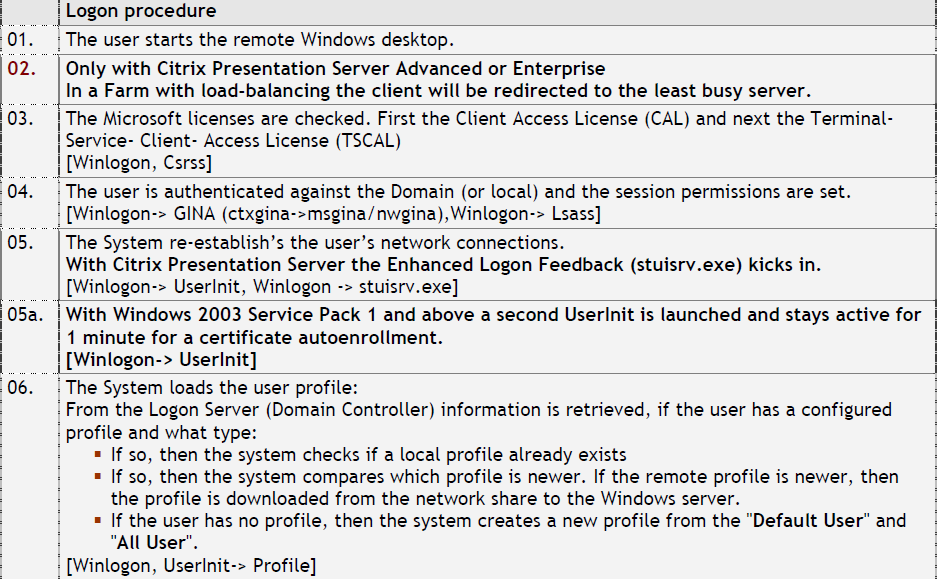
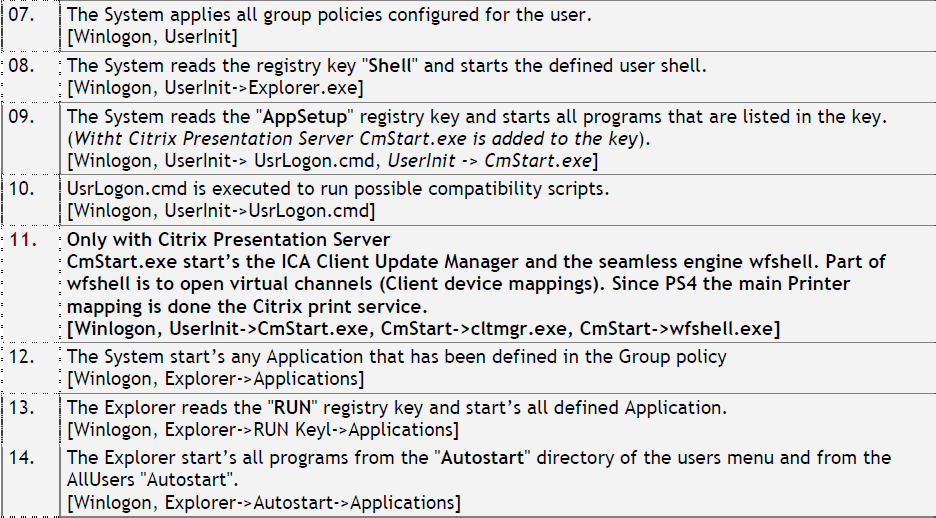
**4. Provides the Interface for Application Software**

All computer operating systems, such as Windows, Unix, and the Mac OS, provide an application program interface for programmers API. API Stands for "Application Program Interface," though it is sometimes referred to as an "Application Programming Interface." An API is a set of commands, functions, and protocols which programmers can use when building software for a specific operating system. The API allows programmers to use predefined functions to interact with the operating system, instead of writing them from scratch.

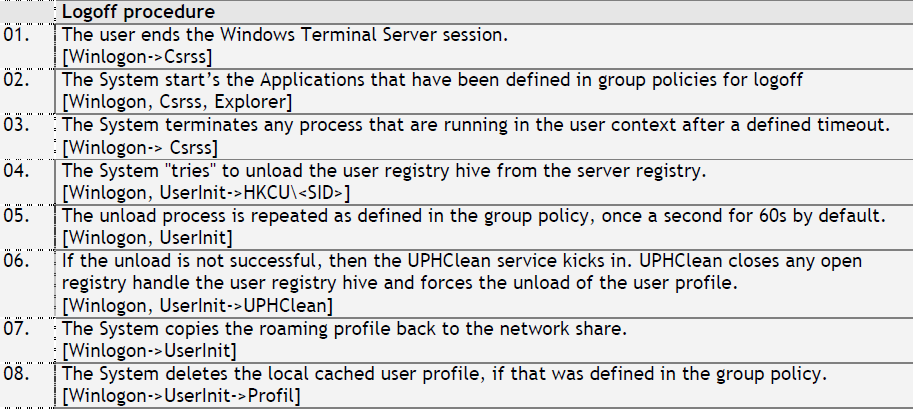
**Q#20 Illustrate the logon And Logout Process in system?**

## Logon

An interactive logon to a computer can be performed either locally, when the user has direct physical access, or remotely, through Terminal Services, in which case the logon is further qualified as remote interactive. After an interactive logon, Windows runs applications on the user’s behalf and the user can interact with those applications.

****

**Logout:-**

In Operating System **Logout/log off** process **is** use to disconnect a remote terminal from a multi-access system by entering (an identification number, password, etc.

**Q#21 Explain different Protection strategies?**

An IT system may need protection for one or more of the following aspects of data:

* **Confidentiality.** The system contains information that requires protection from unauthorized disclosure. Examples: personal information, and proprietary business
* **information. Integrity.** The system contains information that must be protected from unauthorized, unanticipated, or unintentional modification. Examples: economic indicators, or financial transactions systems.
* **Availability.** The system contains information or provides services that must be available on a timely basis to meet mission requirements or to avoid substantial losses. Examples: Systems critical to safety, life support,

**Security Strategies:-**

1. **Data Security:**

* What access controls, integrity controls, and backup procedures are in place to limit attacks?
* Are there privacy policies and procedures that users must comply to?
* What data access controls (authorization, authentication, and implementation) are there?
* What user responsibilities exist for management of data and applications?
* Have direct access storage device management techniques been defined? What is their impact on user file integrity?
* Are there procedures for handling sensitive data?

1. **Network Security:**

* What kinds of access controls (Internet, wide area network connections, etc.) are in place?
* Are there authentication procedures? What authentication protocols are used for local area networks, wide area networks and dialup servers? Who has the responsibility for security administration?
* What type of network media, for example, cables, switches, and routers, are used? What type of security do they have?
* Is security implemented on file and print servers?
* Does your organization make use of encryption and cryptography for use over the Internet, Virtual Private Networks (VPNs), e-mail systems, and remote access?
* Does the organization conform to networking standards?

1. **Physical Security:**

* Are there locks and entry procedures to gain access to servers?
* Is there sufficient air conditioning and are air filters being cleaned out regularly? Are air conditioning ducts safeguarded against break-ins?
* Are there uninterruptible power supplies and generators and are they being checked through maintenance procedures?
* Is there fire suppression and pumping equipment, and proper maintenance procedures for the equipment?
* Is there protection against hardware and software theft? Are software packages and licenses and backups kept in safes?
* Are there procedures for storing data, backups, and licensed software off-site and onsite?

1. **Reactive Strategy**

* A reactive strategy is implemented when the proactive strategy for the attack has failed. The reactive strategy defines the steps that must be taken after or during an attack. It helps to identify the damage that was caused
* that were exploited in the attack, determine why it took place, repair the damage that was caused by it, and implement a contingency plan if one exists.

**Q#22 UNIX File System?**

**Unix file system:-**

The Unix file system (UFS) is a file system used by many Unix and Unix-like operating systems. It is also called the Berkeley Fast File...

A UNIX file system is a collection of files and directories that has the following properties:

* It has a root directory (/) that contains other files and directories.
* Each file or directory is uniquely identified by its name, the directory in which it resides, and a unique identifier, typically called an inode.
* By convention, the root directory has an inode number of 2 and the lost+found directory has an inode number of 3. Inode numbers 0 and 1 are not used. File inode numbers can be seen by specifying the -i option to ls command.
* It is self contained. There are no dependencies between one file system and any other.

A UFS volume is composed of the following parts:

* a few blocks at the beginning of the partition reserved for boot blocks (which must be initialized separately from the filesystem’s)
* a superblock, containing a magic number identifying this as a UFS file system, and some other vital numbers describing this filesystem's geometry and statistics and behavioral tuning parameters
* a collection of cylinder groups. Each cylinder group has the following components:
  + a backup copy of the superblock
  + a cylinder group header, with statistics, free lists, etc., about this cylinder group, similar to those in the superblock
  + a number of inodes, each containing file attributes
  + a number of data blocks

**Q#23 What is System Administrator?**

**System administrator**

|  |
| --- |
| A person in charge of managing and maintaining a computer system of telecommunication system (as for a business or institution). |

### The duties of a system administrator are wide-ranging, and vary widely from one organization to another. Sysadmins are usually charged with installing, supporting and maintaining servers or other computer systems, and planning for and responding to service outages and other problems. Other duties may include scripting or light programming, project management for systems-related projects, supervising or training computer operators, and being the consultant for computer problems beyond the knowledge of technical support staff. To perform his or her job well, a system administrator must demonstrate a blend of technical skills and responsibility.

**The system administrator is responsible for following things:**

1. User administration (setup and maintaining account)
2. Maintaining system
3. Verify that peripherals are working properly
4. Quickly arrange repair for hardware in occasion of hardware failure
5. Monitor system performance
6. Create file systems
7. Install software
8. Create a backup and recovery policy
9. Monitor network communication
10. Update system as soon as new version of OS and application software comes out
11. Implement the policies for the use of the computer system and network
12. Setup security policies for users. A sysadmin must have a strong grasp of computer security (e.g. firewalls and intrusion detection systems).

**system administrator Can:**

* Create new users
* Resetting user passwords
* Lock/unlock user accounts
* Monitor server security
* Monitor special services etc

**Q#24 Differentiate between mounting and un mounting?**

**Mounting:-**

A **mount point** is a physical location in the partition used as a root filesystem. Many different types of storage exist, including magnetic, magneto-optical, optical, and semiconductor (solid-state) drives. Magnetic media are still the most common (as of 2013) and are available as floppy and hard disk drives. Before any of them can be used for storage, the means by which information is read and written must be organized and knowledge of this must be available to the operating system. The organization is called a filesystem. Each different filesystem provides the host operating system with meta data so that it knows how to read and write data. When the medium (or media, when the filesystem is a volume filesystem as in RAID arrays) is mounted, this meta data is read by the operating system so that it can use the storage.

Unix-like operating systems often include software and tools that assist in the mounting process and provide it new functionality. Some of these strategies have been coined "auto-mounting" as a reflection of their purpose.

**Un mounting:-**

A file system is specified by giving the directory where it has been mounted. Giving the special device on which the file system lives may also work, but is obsolete, mainly because it will fail in case this device was mounted on more than one directory. a file system cannot be unmounted when it is `busy' - for example, when there are open files on it, or when some process has its working directory there, or when a swap file on it is in use. The offending process could even be **umount** itself - it opens libc, and libc in its turn may open for example locale files. A lazy unmount avoids this problem.

Options for the **umount** command:

**-V**

Print version and exit.

**-h**

Print help message and exit.

**-v**

Verbose mode.

**-n**

Unmount without writing in */etc/mtab*.

**-r**

In case unmounting fails, try to remount read-only.

**-d**

In case the unmounted device was a loop device, also free this loop device.

**-a**

All of the file systems described in */etc/mtab* are unmounted. (With **umount** version 2.7 and later: the *proc* filesystem is not unmounted.)

**Q#25 Define UNIX Shell?**

**UNIX Shell**

A **Unix shell** is a command-line interpreter or shell that provides a traditional user interface for the Unix operating system and for Unix-like systems. Users direct the operation of the computer by entering commands as text for a command line interpreter to execute or by creating text scripts of one or more such commands. The most influential Unix shells have been the Bourne shell and the C shell.

**Bourne shell:-**

The Bourne shell, sh, was written by [Stephen Bourne](http://en.wikipedia.org/wiki/Stephen_Bourne) at AT&T as the original Unix command line interpreter; it introduced the basic features common to all the Unix shells

**C shell:-**

The C shell, csh, was written by Bill Joy while a graduate student at University of California, Berkeley. The language, including the control structures and the expression grammar, was modeled on C. The C shell also introduced a large number of features for interactive work, including the history and editing mechanisms, aliases, directory stacks, tilde notation, cdpath, job control and path hashing.

**Shell commands:-**

Shell commands vary from shell to shell (bash shell, borne shell, C shell, korn shell etc), but remain the same across Linux distributions. As with the Linux commands, we have compiled a list of bash shell commands, and are working our way through writing the basic explanations for those commands

**Q#26 Explain function of shell?**

**Shell functions**

A shell function is a user-defined name that acts as a simple command. It is defined with a function definition command, and may be called in the same way as any other command.

**Arguments**

A function can receive arguments on the command line just as any other command, and they are accessed in the same way — as positional parameters.

**Function definition command**

A function is defined with its name followed by an empty pair of parentheses, **()**, then a compound command and redirection (if any). The compound command is referred to as the body of the function.

**Syntax**

Functions are declared using this syntax:

name () compound-command [redirections]

or

function name [()] compound-command [redirections]

Examples

pr1 () ## print each argument on a separate line

{

printf "%s\n" "$@"

}

is\_int ()

case $1 in

\*[^0-9]\*) return 1;; ## not a positive integer

esac

If the compound command is enclosed in a pair of parentheses, it will be executed in a subshell, and any variable definitions or modifications in the body of the function will not be visible to the calling process.

**Q# 27 Explain Vi Editor?**

**VI Editor:-**

The vi editor (short for visual editor) is a screen editor which is available on almost all Unix systems.vi has no menus but instead uses combinations of keystrokes in order to accomplish commands.

VI is available by default on every Unix machine Because of its universal availability.

Different vi clones of course have different ways of starting the program (invocation). Usually, however, command-line versions of vi share a common basic set of command line options. These following command line options and flags are typically available. In addition, vi can be started under different names. Depending on the name used to start vi, it may either behave slightly differently or load a different vi clone.

The common command line options and flags are

-

or

-s

**Q# 28 Describe the different modes of VI Editor?**

**VI Editor Modes**

vi has 3 modes:

1. write mode, used for entering text
2. command mode and
3. command line mode, used for entering commands to vi.

**Write Mode**

When you first enter the editor, you are in the command mode. To enter the write mode, type the letter a for append. This is one of the four possible commands for entering the write mode. vi is Case Sensitive. Lower case commands are different from upper case commands.

**Command Mode**

You are in command mode whenever you hit esc to leave the write mode. In command mode, you can move the cursor anywhere in the file.

**Command-Line Mode**

Command-line mode is used for such things as writing changes and exiting the editor. To enter command-line mode, type : while in command mode. The : will now appear at the bottom of the screen and the command which you type will appear on that line

**Q# Write Short Note :**

1. **Input and Output devices**
2. **Storage devices**
3. **FTP protocol**
4. **Virtual memory**
5. **Input and Out Put Devices:-**

**Input Devices:-**

**(a) Keyboard**

It is a text base input device that allows the user to input alphabets, numbers and

Other

**(b) Mouse**

The mouse is a small device used to point to a particular place on the screen and

select in order to perform one or more actions. It can be used to select menu

commands, size windows, start programs etc.

**(c) Joystick**

The joystick is a vertical stick which moves the graphic cursor in a direction the stick

is moved. It typically has a button on top that is used to select the option pointed by

the cursor. Joystick is used as an input device primarily used with video games,

training simulators and controlling robots

**(d)Scanner**

Scanner is an input device used for direct data entry from the source document into

the computer system. It converts the document image into digital form so that it can

be fed into the computer.

**(e) Light Pen**

It is a pen shaped device used to select objects on a display screen. It is quite like the

mouse (in its functionality) but uses a light pen to move the pointer and select any

object on the screen by pointing to the object.

**(f) Touch Screen**

It allows the user to operate/make selections by simply touching the display screen.

Common examples of touch screen include information kiosks, and bank ATMs.

**(g)Digital camera**

A digital camera can store many more pictures than an ordinary camera. Pictures

taken using a digital camera are stored inside its memory and can be transferred to a

computer by connecting the camera to it.

**Output Devices**

**(a) Monitor**

Monitor is an output device that resembles the television screen and uses a Cathode

Ray Tube (CRT) to display information. The monitor is associated with a keyboard

for manual input of characters and displays the information as it is keyed in. It also

displays the program or application output.

**(b) Liquid Crystal Display (LCD)**

LCD was introduced in the 1970s and is now applied to display terminals also. Its

advantages like low energy consumption, smaller and lighter have paved its way for

usage in portable computers (laptops).

**(c) Printer**

Printers are used to produce paper (commonly known as hardcopy) output. Based on

the technology used, they can be classified as Impact or Non-impact printers.

**(d) Plotter**

Plotters are used to print graphical output on paper. It interprets computer commands

and makes line drawings on paper using multicolored automated pens. It is capable of

producing graphs, drawings, charts, maps etc.

**(e) Audio Output: Sound Cards and Speakers:**

The Audio output is the ability of the computer to output sound. Two components are

needed: Sound card – Plays contents of digitized recordings, Speakers – Attached to

sound card.

1. **Storage Devices:-**

Physical components or materials on which data is stored are called storage media.

Hardware components that read/write to storage media are called storage devices.

Two main categories of storage technology used today are magnetic storage and

optical storage.

**Primary magnetic storage**

o Diskettes

o Hard disks (both fixed and removable)

o High capacity floppy disks

o Disk cartridges

o Magnetic tape

**Primary optical storage**

o Compact Disk Read Only Memory (CD ROM)

o Digital Video Disk Read Only Memory (DVD ROM)

o CD Recordable (CD R)

o CD Rewritable (CD RW)

o Photo CD

**uses for disks**

· Moving files between computers not connected by a network or

communication hardware – one of the easiest ways to move data from one

computer to another is to copy it onto diskette and use the diskette in another

computer.

· Loading new programs onto a system – Large programs are usually delivered

by CD-ROM but many programs are still sold on diskette. You install it by

copying the contents of the disks onto the hard drive and then running a small

program, which installs the files automatically.

· Backing up data or programs – Primary copy of data or programs are stored

on the hard drive, Backing up is the process of creating a duplicate copy for

safekeeping. To protect against data loss it is wise to back up a hard disk.

Due to the limited storage capacity diskettes are used to back up small data

files rather than programs or entire hard disks.

**Hard Disks**

Hard disk is still the most common storage device for all computers. Like diskettes

hard drives store data in tracks divided into sectors. Physically however they look

quite different to diskettes.

**Removable disks**

· High capacity floppy disks

Disks with the same dimensions as 3.3-inch floppy diskette or slightly larger

have a much greater capacity than a standard diskette.

**Optical Storage Devices**

The most popular alternative to magnetic storage systems are optical storage media.

The most widely used type of optical storage medium is the compact disk (CD),

this is used in CD-ROM, DVD-ROM, CDR, CDRW and PhotoCD systems.

1. **FTP(File Transfer Protocol)**

**File Transfer Protocol** (**FTP**) is a standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet.

FTP is built on a client-server architecture and uses separate control and data connections between the client and the server

The first FTP client applications were command-line applications developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems.

Data transfer can be done in any of three modes:

**Stream mode:** Data is sent as a continuous stream, relieving FTP from doing any processing. Rather, all processing is left up to TCP. No End-of-file indicator is needed, unless the data is divided into records.

* **Block mode:** FTP breaks the data into several blocks (block header, byte count, and data field) and then passes it on to TCP.
* **Compressed mode:** Data is compressed using a single algorithm (usually run-length encoding)

**Security**

FTP was not designed to be a secure protocol—especially by today's standards—and has many security weaknesses which are follows:

* Brute force attacks
* Bounce attacks
* Packet capture (sniffing)
* Port stealing
* Spoof attacks
* Username protection

1. **Virtual Memory**

In Operating System **virtual memory** is a memory management technique developed for multitasking kernels. This technique virtualizes the main storage available to a process or task, as a contiguous address space which is unique to each running process, or virtualizes the main storage available to all processes or tasks on the system as a contiguous global address space. The operating system manages virtual address spaces and the assignment of real memory to virtual memory. The CPU automatically translates virtual addresses to physical addresses using special memory management hardware, often referred to as a memory management unit.

**Usage**

Virtual memory is an integral part of a modern computer architecture; implementations require hardware support, typically in the form of a memory management unit built into the CPU. While not necessary, emulators and virtual machines can employ hardware support to increase performance of their virtual memory implementations.

**Q#30What is Operating System? Differentiate b/w command mode and Graphical Mode system?**

**Operating System**

The 1960’s definition of an operating system is “the software that controls the hardware”. However, today, due to microcode we need a better definition. We see an operating system as the programs that make the hardware useable. In brief, an operating system is the set of programs that controls a computer. Some examples of operating systems are UNIX, Mach, MS-DOS, MS-Windows, Windows/NT, Chicago, OS/2, MacOS, VMS, MVS, and VM.

Operating Systems are resource managers. The main resource is computer hardware in the form of processors, storage, input/output devices, communication devices, and data. Some of the operating system functions are: implementing the user interface, sharing hardware among users, allowing users to share data among themselves, preventing users from interfering with one another, scheduling resources among users, facilitating input/output, recovering from errors, accounting for resource usage, facilitating parallel operations, organizing data for secure and rapid access, and handling network communications.

Differentiate b/w Command mode and Graphical Mode system:

**COMMAND Mode:-**

COMMAND mode is the filename of the default operating system shell for DOS operating systems and the default command line interpreter on Windows 95, Windows 98 and Windows Me. It has an additional role as the first program run after boot, hence being responsible for setting up the system by running the AUTOEXEC.BAT configuration file, and being the ancestor of all processes. COMMAND's successor on OS/2 and Windows NT-based operating systems is cmd.exe (Command Prompt). COMMAND is also available on 32-bit versions of those systems to provide compatibility when running DOS applications within the NT Virtual DOS machine

**Grafical Mode:-**

A **graphical model** is a probabilistic model for which a graph denotes the conditional dependence structure between random variables. They are commonly used in probability theory, statistics—particularly Bayesian statistics—and machine learning.

**Types of graphical models**

Generally, probabilistic graphical models use a graph-based representation as the foundation for encoding a complete distribution over a multi-dimensional space and a graph that is a compact or factorized representation of a set of independences that hold in the specific distribution. Two branches of graphical representations of distributions are commonly used, namely,

1. **Bayesian networks**
2. **Markov networks**.

Q#31 what is system software?

# System software

**System software** is an operating system designed to operate and control the computer hardware and to provide a platform for running application software. Device drivers such as computer BIOS and device firmware provide basic functionality to operate and control the hardware connected to or built into the computer. The operating system (prominent examples being z/OS, Microsoft Windows, Mac OS X and Linux), allows the parts of a computer to work together by performing tasks like transferring data between memory and disks or rendering output onto a display device. It also provides a platform to run high-level system software and application software.

In some publications, the term *system software* also includes software development tools (like a compiler, linker or debugger). In contrast to system software, software that allows users to do things like create text documents, play games, listen to music, or surf the web is called application software

**Q#32 what do you by permissions or rights of multi user environment?**

**Multiuser Environments**

A multiuser environment is one in which other users can connect and make changes to the same database that you are working with. As a result, several users might be working with the same database objects at the same time. Thus, a multiuser environment introduces the possibility of the database being affected by changes made by other users while you are making changes

A multi user virtual environment is created in three steps. The first is a server or a farm of servers, which are used as the host of the virtual world. Second, a program or an interface is needed that allows people to create a user name and some sort of identity that they can use when they log into the server. The third is there has to be some reason for the person to want to be in the Virtual Environment.

**Permissions or rights of multi user environment:**

The multi-user environment provides read/write access control and tracking for two or more users working with the same files concurrently. Users share a file when they have loaded the same file from the same location.

A key issue when working with databases in a multiuser environment is access permissions. The permissions you have for the database determine the extent of the work you can do with the database. For example, to make changes to objects in a database, you must have the appropriate write permissions for the database. For more information about permissions in your database, see your database documentation.

Q#33 What is Backup and Recovery?

# Backup

In information technology, a **backup**, or the process of **backing up**, refers to the copying and archiving of computer data so it may be used to *restore* the original after a data loss event. The verb form is **to back up** in two words, whereas the noun is *backup*. Backups have two distinct purposes. The primary purpose is to recover data after its loss, be it by data deletion or corruption. Data loss can be a common experience of computer users. A 2008 survey found that 66% of respondents had lost files on their home PC.The secondary purpose of backups is to recover data from an earlier time, according to a user-defined data retention policy, typically configured within a backup application for how long copies of data are required. Though backups popularly represent a simple form of disaster recovery, and should be part of a disaster recovery plan, by themselves, backups should not alone be considered disaster recovery

**Recovery:-**

A **recovery** is a general term for media containing a backup of the original factory condition or a favored condition of a computer as configured by an original equipment manufacturer or an end-user. OEM supplied recovery media is commonly shipped with most computers to allow the user to reformat the hard drive and reinstall the operating system and pre-loaded software as it was when it was shipped.

**Recovery partitions**

In recent years, bundled recovery CD-ROMs have become less common, as some OEM's are now utilizing hard drive partitions to store the recovery data. Accessing hard drive based system recovery is usually performed by pressing a specific key combination during or after the computer's POST.

**Q#34 What is protocol? Explain TCP/IP and FTP protocol?**

**Protocol:-**

These set of rules are called protocols. Imagine the number of people communicating in the world, the number of different languages they use, the number of different machines they use, the number of ways in which they transmit data and the different software they use. We would never be able to communicate worldwide if there were no ‘standards’ governing the way we communicate and the way our machines treat data. These standards are sets of rules.

There are many protocols, each one governing the way a certain technology works

1. **TCP/IP Protocol**
2. **FTP Protocol**

**TCP protocol:-**

Transmission Control Protocol, used for the reliable transmission of data over a network.

he **Transmission Control Protocol** (**TCP**) is one of the two original core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called *TCP/IP*. TCP provides reliable, ordered, error-checked delivery of a stream of octets between programs running on computers connected to an intranet or the public Internet.

**FTP protocol:-**

**File Transfer Protocol** (**FTP**) is a standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet.

FTP is built on client-server architecture and uses separate control and data connections between the client and the server.FTP users may authenticate themselves using a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that hides (encrypts) the username and password, and encrypts the content, FTP is often secured with SSL/TLS ("FTPS"). SSH File Transfer Protocol ("SFTP") is sometimes also used instead, but is technologically different.

FTP is the easiest way to transfer files between computers via the internet, and utilizes TCP, transmission control protocol, and IP, internet protocol, systems to perform uploading and downloading tasks.

**Q#35 Define layer approach of operating system?**

**Layered Approach:**

Layered Approach **With** proper hardware support, operating systems can be broken into pieces that are smaller and more appropriate than those allowed by the original MS-DOS or UNIX systems. The operating system can then retain much greater control over the computer and over the applications that make use of that computer. Implementers have more freedom in changing the inner workings of the system and in creating modular operating systems. Under the top-down approach, the overall functionality and features are determined and the separated into components. Information hiding is also important, because it leaves programmers free to implement the low-level routines as they see fit, provided that the external interface of the routine stays unchanged and that the routine itself performs the advertised task.

Q#36 Explain in detail the term Main Memory?

**Main Memory**

The main memory stores data and instructions. Main memories are usually built from dynamic IC’s known as dynamic RAMs. These semiconductor ICs can also implement static memories referred to as static RAMs (SRAMs). SRAMs are faster but cost per bit is higher. These are often used to build caches.

**Types of Random-Access Semiconductor Memory**

Dynamic RAM: for example: Charge in capacitor. It requires periodic refreshing. Static RAM: for example: Flip-flop logic-gate. Applying power is enough no need for refreshing). Dynamic RAM is simpler and hence smaller than the static RAM. Therefore more dense and less expensive. But it requires supporting refresh circuitry. Static RAM/s are faster than dynamic RAM’s.

ROM: The data is actually wired in the factory. Can never be altered.

PROM: Programmable ROM. It can only be programmed once after its fabrication. It requires special device to program.

EPROM: Erasable Programmable ROM. It can be programmed multiple times. Whole capacity need to be erased by ultraviolet radiation before a new programming activity. It can be partially programmed.

EEPROM: Electrically Erasable Programmable ROM. Erased and programmed electrically. It can be partially programmed. Write operation takes considerably longer time compared to read operation.

Each more functional Rom is more expensive to build, and has smaller capacity then less functional ROM’s.

**Q#37 Principles of Unix/Linux Operating System?**

An American programmer and open source advocate, summarizes the UNIX philosophy as KISS Principle of "Keep it Simple, Stupid He provides a series of design rules

* Rule of Modularity: Developers should build a program out of simple parts connected by well defined interfaces, so problems are local, and parts of the program can be replaced in future versions to support new features. This rule aims to save time on debugging complex code that is complex, long, and unreadable.
* Rule of Clarity: Developers should write programs as if the most important communication is to the developers, including him- or herself, whom will read and maintain the program rather than the computer. This rule aims to make code readable and comprehensible for whom ever works on the code in future.
* Rule of Composition: Developers should write programs that can communicate easily with other programs. This rule aims to allow developers to break down projects into small, simple programs rather than overly complex monolithic programs.
* Rule of Separation: Developers should separate the mechanisms of the programs from the policies of the programs; one method is to divide a program into a front-end interface and back-end engine that interface communicates with. This rule aims to let policies be changed without destabilizing mechanisms and consequently reducing the number of bugs.
* Rule of Simplicity: Developers should design for simplicity by looking for ways to break up program systems into small, straightforward cooperating pieces. This rule aims to discourage developers’ affection for writing “intricate and beautiful complexities” that are in reality bug prone programs.
* Rule of Parsimony: Developers should avoid writing big programs. This rule aims to prevent overinvestment of development time in failed or suboptimal approaches caused by the owners of the program’s reluctance to throw away visibly large pieces of work. Smaller programs are not only easier to optimize and maintain; they are easier to delete when deprecated.
* Rule of Transparency: Developers should design for visibility and discoverability by writing in a way that their thought process can lucidly be seen by future developers working on the project and using input and output formats that make it easy to identify valid input and correct output. This rule aims to reduce debugging time and extend the lifespan of programs.
* Rule of Robustness: Developers should design robust programs by designing for transparency and discoverability, because code that is easy to understand is easier to stress test for unexpected conditions that may not be foreseeable in complex programs. This rule aims to help developers build robust, reliable products.
* Rule of Representation: Developers should choose to make data more complicated rather than the procedural logic of the program when faced with the choice, because it is easier for humans to understand complex data compared with complex logic. This rule aims to make programs more readable for any developer working on the project, which allows the program to be maintained.
* Rule of Least Surprise: Developers should design programs that build on top of the potentials users' expected knowledge; for example, ‘+’ should always mean addition in a calculator program. This rule aims to encourage developers to build intuitive products that are easy to use.
* Rule of Silence: Developers should design programs so that they do not print unnecessary output. This rule aims to allows other programs and developers to pick out the information they need from a program's output without having to parse verbosity.
* Rule of Repair: Developers should design programs that fail in a manner that is easy to localize and diagnose or in other words “fail noisily”. This rule aims to prevent incorrect output from a program from becoming an input and corrupting the output of other code undetected.
* Rule of Economy: Developers should value developer time over machine time, because machine cycles as of the year 2013 are relatively inexpensive compared to prices in the 1970s. This rule aims to reduce development costs of projects.
* Rule of Generation: Developers should avoid writing code by hand and instead write abstract high-level programs that generate code. This rule aims to reduce humans errors and save time.
* Rule of Optimization: Developers should prototype software before polishing it. This rule aims to prevent developers from spending too much time for marginal gains.
* Rule of Diversity: Developers should design their programs to be flexible and open. This rule aims to make programs flexible, allowing them to be used in other ways than their developers intended.
* Rule of Extensibility: Developers should design for the future by making their protocols extensible, allowing for easy plug-in without modification to the program's architecture by other developers, noting the version of the program, and more. This rule aims to extend the lifespan and enhance the utility of the code the developer writes.

**Q#38 Command Of following Topics:**

1. **Process related Command**In most Unix-like operating systems, the **ps** program (short for "**p**rocess **s**tatus") displays the currently-running processes. A related Unix utility named top provides a real-time view of the running processes.

In Windows PowerShell, ps is a predefined command alias for the Get-Process cmdlet which essentially serves the same purpose.

**Examples**

# ps

PID TTY TIME CMD

7431 pts/0 00:00:00 su

7434 pts/0 00:00:00 bash

18585 pts/0 00:00:00 ps

1. **File and directory Command:**

**File Command:file** is a standard Unix program for recognizing the type of data contained in a computer

**Examples**

# file file.c

file.c: C program text

# file program

program: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked

(uses shared libs), stripped

**Directory Command:** The term *directory* is used in a computer software context to refer to what appears to the user to be a *container* or *folder*1 that can hold *files* and other directories

**Example:**

mkdir [name] (CR)

pushd [name] (CR)

popd (cd)

cd [directory] (CR)

**Compress Command**

Compacts a file so that it is smaller. When compressing a file it will be replaced with a file with the extension .Z, while keeping all the same ownership modes.

**Syntax**

*compress [-c] [-f] [-v] filenames*

Examples

**compress -v bigfile.exe**

Compress bigfile.exe and rename that file to bigfile.exe.Z.

**Uncompress Command:-**

Restores compressed files.

**Syntax: uncompress** [  [**-c**](http://pic.dhe.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.cmds/doc/aixcmds5/uncompress.htm#uncompress__row-d3e145644)] [  [**-F**](http://pic.dhe.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.cmds/doc/aixcmds5/uncompress.htm#uncompress__row-d3e145650)] [  [**-f**](http://pic.dhe.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.cmds/doc/aixcmds5/uncompress.htm#uncompress__row-d3e145650)] [  [**-n**](http://pic.dhe.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.cmds/doc/aixcmds5/uncompress.htm#uncompress__row-d3e145665)] [  [**-q**](http://pic.dhe.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.cmds/doc/aixcmds5/uncompress.htm#uncompress__row-d3e145677)] [  [**-V**](http://pic.dhe.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.cmds/doc/aixcmds5/uncompress.htm#uncompress__row-d3e145692)] [ [File ...](http://pic.dhe.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.cmds/doc/aixcmds5/uncompress.htm#uncompress__row-d3e145710) ]

**Example:** To uncompress the foo.Z file, enter:

uncompress foo.Z

The foo.Z file is uncompressed and renamed foo.

**Cpio Command**

Creates and un-creates archived cpio files. And also is capable of copying files to things other than a hard disk.

**Syntax:** *cpio -i [ bBcdfkmPrsStuvV6 ] [ -C bufsize ] [ -E file ] [ -H header ] [ -I file [ -M message ] ] [ -R id ] [pattern ... ]*

*cpio -o [ aABcLPvV ] [ -C bufsize ] [ -H header ] [ -O file [ -M message ] ]*

*cpio -p [ adlLmPuvV ] [ -R id ] directory*

**Examples:** find . -print | cpio -ocv > /dev/fd0

find . -print | cpio -dumpv /home/users/hopecpio -icuvd < /dev/fd0

**Manual Command**

The man command is short for manual and provides in depth information about the requested command or allows users to search for commands related to a particular keyword.

**Syntax:** *man [-] [-k keywords] topic*

**Examples:** man mkdir

Lists help information on the mkdir command.

man -k irc

**Job command**

Lists the jobs that you are running in the background and in the foreground. If the prompt is returned with no information no jobs are present. Note: not all shells are capable of running this command.

**Syntax:** *jobs [-p | -l] [-n] [-p] [-x] [job id]*

**Examples:** jobs

Would display results similar to the below if jobs were running in the background.

[1] + Stopped (user)      man jobs

jobs -l

The above command would not just list the jobs running but also this group ID and the working directory of the jobs.

**Fold Command**

Filter for folding lines. This breaks the lines to have a maximum of x width column position (or bytes).

**Syntax:** *fold [ -bs ] [-w width | -width] [file]*

**Examples** fold -5 myfile.txt > newfile.txt

In the above command, this would fold the lines of myfile.txt to 5 character width and re-route the results to the file newfile.txt