**Working with Windows/Unix install basic concept**

**Installing an Operating System**

* Hardware Requirements - Before you attempt to install an OS on a machine, it would be wise to ensure that the OS in question runs on the hardware that you have available.
* Multi-platform OS Support - Most OS vendors that distribute software that runs on multiple hardware architectures publish a hardware compatibility list (HCL) for their OS.
* Single Platform OS Support - Vendors of single-architecture OS deal with many of the same problems faced by multi-platform vendors.

**OS for Intel Based**

Windows is probably the most widely used OS available for Intelbased systems.

* Windows includes drivers for a wide range of devices that work on Intel-based systems.
* Versions of Windows produced over the past decade will operate on systems ranging from a 486 chip up to the latest Pentium.
* The current versions of Windows generally require a Pentium 166 (or better) processor chip, a minimum of 32 megabytes of memory, and 2 gigabytes (or more) of free disk space.
* Windows may not operate smoothly on some of the older systems, but it will operate.

Recent releases of Linux are beginning to nip at the heels of Windows in regard to device driver availability, and the number of systems running the OS.

* The open-source movement has the advantage of allowing anyone to submit a device driver for their favorite device.

Because there are so many PC systems available, many people know how to write “bad” software for the PC architecture.

* This plethora of untested/unsupported software available for the system can lead to security and reliability problems. Both Linux, and Windows suffer because of this situation.

Well behind Windows and Linux, BSD, and Solaris are also available for Intel architecture systems.

* Solaris suffers from a lack of device drivers for the latest/greatest devices.
* Solaris also often does not offer device drivers for very old devices.
* Sun discontinued support for the Solaris Intel edition with the release of Solaris 9. They now offer a Linux “solution” in place of Solaris for Intel.
* BSD variants are generally more secure than the competitors, and offer very robust/well tested code, but they have not caught on as general-purpose offerings. BSD is often found in research labs, and under the hood of network “appliances” such as firewalls.

**OS for Non-Intel-based Systems**

Proprietary hardware architectures may not allow you to run more than one OS.

* The “single offering” case greatly simplifies the decision regarding which OS you should load on the machine.
* The single OS is also easier for the vendor to support, as the service personnel only have one system to learn.
* Single-OS systems are also likely to be replaced by multi-OS systems over time.
* The world of computing is not a “one-size-fits-all” environment.
* The UNIX vs. Windows war will be battled for many years.

Whereas some architectures may only work with one OS, others are blessed with multiple fully supported OS.

* Some vendors may provide full support for their proprietary OS but partial support for a second OS.
* More often than not, the second OS is one of the Linux distributions.
* Vendors such as Hewlett-Packard, IBM, Compaq and others now offer multiple OS for their hardware.
* Other vendors (like Sun) offer a line of Intel based Linux systems, as well as the SPARC Scalable Processor Architecture based systems.

**Cataloging Hardware Requirement**

When you consider the OS hardware requirements, you need to pay close attention to any nonstandard hardware on your system.

All hardware found on the system should be cataloged.

* Need to determine if device drivers are available.
* Need to determine if the hardware is mission critical.

Looking over the inventory form before attempting to install the latest OS version may make the decision for you.

PC systems typically force a few more restraints on the system hardware than other architectures.

* For example, an Intel system requires a VGA display and keyboard in order to boot.
* Unfortunately, there are hundreds of display adapters, thousands of keyboards, and tens of thousands of mice available for PCs of different heritage.

OS designers also impose restraints on the system hardware.

* Software kernel architectures may impose specific revision-level requirements on the hardware.
* For example, Solaris 8 will not boot on the sun4c hardware architecture.
* For example, Solaris 8 will not boot on the sun4c hardware architecture.

**Installation Time Requirement**

* When it comes to a discussion of installing an OS, the answer to “what hardware do I need?” is almost as nebulous as the answer to “how long will it take?”
* When everything goes right, a full OS install might take an hour. But when things go wrong, you might be looking at eight or more hours to get the OS installed and working.
* Never assume the OS installation will go smoothly.
* Even if you get extremely lucky, and everything does go well, figure on at least two hours just to get a bootable OS on the system media.
* Do not forget that after the OS is installed you will need to install patches and service packs to bring the OS up to date.
* Once the OS is installed and patched, do not forget that you have to install all of the applications users need in order to get their work done.
* Allow time for the appropriate number of reboots during the OS and application installation phases.
* The “load” media for an install can also have a huge impact on the installation.

**Types of Installations**

**There are several types of installation models available to work with.**

* The installation could be a server or a workstation.
* The goal of the installation could be to set up a machine usable as a desktop workstation, web server, database server, service machine, or interactive system.
* The system could be a diskless host that boots from the network, or a diskful system that serves files to other hosts.
* Each of these configurations may require that a particular set of software be installed such that the system performs its duties and users can get some work done.

**Some of the more common problems encountered while installing Windows include:**

**One or more network interfaces may not operate correctly** - Generally, removing the adapter’s driver, rebooting the system, and reinstalling the driver will correct this problem.

**NT 4.0 cannot be installed on systems with a boot partition larger than 4 gigabytes -** This limit requires you to partition system disks larger than 4 gigabytes into (at least) two partitions.

**The device drivers on the Windows distribution media are often less than optimal -** Display and network drivers seem to be the “big losers” in this respect.

**Service packs have been issued for all current versions of Windows -** It is imperative that the sysadmin install these service packs to secure the system against well-known attacks.

**Dual-boot Windows Installations**

* The idea behind a dual-boot system is to install two different OS on the bootable medium.
* This allows the user to run either system, which hopefully allows the user more flexibility in performing tasks.
* Unfortunately, creating a dual-boot system is a lot more difficult than it should be, and many times Windows seems to be in the middle of the problems.
* One rule of thumb for installing a dual-boot system that includes Windows is to install the Windows OS first.
* If you are trying to install two versions of Windows, always install the older version first.
* This is often necessary because new versions of Windows often contain changes to the file system, and the old versions do not understand these changes.

**Other Installations**

* Installing a Solaris/Linux dual-boot Intel system often presents problems for the administrator.
* The Solaris Intel installer limits the number of partitions allowed on a disk where Solaris will be installed.
* If the disk has more than three partitions, and/or the installation of Solaris would cause the disk to have more than three partitions, the installation will fail.
* Linux wants a boot partition, and a kernel partition (minimum).
* Solaris also wants a boot partition and a kernel partition.
* If you wanted to build a dual-boot Linux/Solaris system, you would need four partitions. However, Solaris will only allow you to install on a disk with three (or fewer) partitions.
* In this case, install Solaris first, and then install Linux on a separate partition later. Better yet, buy a second disk, and install one OS on each disk.

**Desktop Installation**

**Installing the OS on a desktop PC is often a very different problem than installing an OS on a corporate database server** - Generally, desktop computers come in two flavors: systems that contain their own copies of everything and systems that rely on network-based servers for critical applications.

**Standalone Systems** - Self-contained systems are often referred to as standalone systems, or “thick clients.

* These machines typically contain the OS, and local copies of all applications required by users of the system.
* The installation of a standalone system will require more time than some other systems because you have to load the OS, and all of the applications on the local disk.
* Such installations can easily become multi-day tasks

**Server Installations**

* Installing an OS on a server is often a long, arduous task.
* You have to install the OS and configure it to provide services to other computers.
* The types of “clients” the server supports will usually complicate this configuration task.
* The applications/services provided by the server may provide more complications.

**Homogenous Servers**

* The homogenous server is probably the simplest server to install.
* This type of server only provides services to clients of the same architecture/kernel architecture.
* This means that only one version of the OS and all applications need be installed on the system.
* Such systems may be used as a boot server, file server, name server, web server, database server, or many other purposes

**Heterogenous Servers**

* Heterogeneous servers are probably the most difficult system you will ever have to install.
* These systems may provide boot services, applications, and/or file services for a variety of systems of different kernel/hardware architectures.
* For example, a Linux system may be set up to provide file service to Linux, Solaris, and MacOS boxes via NFS, while providing file service to desktop PCs via Common Internet File Services (CIFS) by running the Samba application. Such servers are typically very complicated beasts to install and configure.
* You will have to install copies of multiple OS for the system to function as a boot server.
* Similarly, you will have to install application binaries for multiple architectures in order to support application services for client machines

**Planning for an Installation**

* The “footprint” or size of the OS should be considered to ensure that the system contains enough disk space.
* How that disk space is parceled might play a role in the OS installation.
* Similarly the size of the main memory might need to be taken into consideration.

**Disk Space Requirements**

* One of the most important decisions you will need to make before you install an OS is how much space to allocate to the system software.
* If you allocate too much space to the system software, users may not have enough space. If you allocate too much space to users, the system may run out of space.
* Calculate how much space the OS and application binaries will occupy.
* Once you have a number in mind, double it. In a few weeks or months, you will be glad you did.

**Every successive release of an OS is larger than its predecessor -** This is an important point to remember, because you may have to upgrade the OS on a machine two or three times over the lifetime of the hardware.

**Operating System Installation: Initial Setup and Patching**

1. System security begins with the installation of the os.
2. Ideally new systems should be constructed on a protected network.
3. The initial installation should compromise the minimum necessary for the desired system, with additional software packages included only if they are required for the function of the system.
4. The overall boot process must also be secured.
5. Care is also required with the selection and installation of any additional device driver code, since this executes with full kernel level privileges, but is often supplied by a third party.

**Operating systems hardening**

Systems hardening is a collection of tools, techniques, and best practices to reduce vulnerability in technology applications, systems, infrastructure, firmware, and other areas.

* OS installation: initial setup and patching
* remove unnecessary services, application, and protocols
* configure users, groups and authentication
* install additional security controls
* test the system security

**Methods of Installing a new computer-based system**

**Direct Changeover** - New system completely replaces the old without overlap

**Advantages:**

* –changeover is swift (no time wasted)
* only one system needs to be supported
* minimum duplication of work –minimum disruption to the business

**Disadvantages:**

* operations disrupted if the new system has errors
* loss of confidence if things go wrong
* staff stress/confidence/training.

**Parallel Changeover** - Old system continues alongside the new system –for a period of time

**Advantages:**

* new system can be checked against results from old system,
* operations can continue while problems sorted out,
* users can learn the new system at their own pace.

**Disadvantages:**

* duplication of effort required to keep systems running, may put a strain on personnel.

**Phased Changeover** - Installation broken down into individual modules installed separately, at different times

**Advantages:**

* installation, training, and costs can be spread over time
* faults can be corrected with limited effect on the business

**Disadvantages:**

* time taken for full installation slow to commission the whole system

**Pilot Changeover** - Introduced to an area of the organization for a given time. –maybe a department or branch. If system works correctly, it is activated at the next branch

**Advantages:**

* results can be compared with the old system, bugs can be cleared from the system,
* before full roll-out training can be modified in the light of experience.

**Disadvantages:**

* complete roll-out can take some time

**Benefits of Systems Hardening**

* Enhanced system functionality: Since fewer programs and less functionality means there is less risk of operational issues, misconfigurations, incompatibilities, and compromise.
* Significantly improved security: A reduced attack surface translates into a lower risk of data breaches, unauthorized access, systems hacking, or malware.
* Simplified compliance and auditability: Fewer programs and accounts coupled with a less complex environment means auditing the environment will usually be more transparent and straightforward

**Systems Hardening to Reduce the “Attack Surface”**

The “attack surface” is the combination of all the potential flaws and backdoors in technology that can be exploited by hackers. These vulnerabilities can occur in multiple ways, including:

* Default and hardcoded passwords
* Passwords and other credentials stored in plain text files
* Unpatched software and firmware vulnerabilities
* Poorly configured BIOS, firewalls, ports, servers, switches, routers, or other parts of the infrastructure
* Unencrypted network traffic or data at rest
* Lack, or deficiency, of privileged access controls

**9 Best Practices for Systems Hardening**

1. **Audit your existing systems:** Carry out a comprehensive audit of your existing technology. Use penetration testing, vulnerability scanning, configuration management, and other security auditing tools to find flaws in the system and prioritize fixes. Conduct system hardening assessments against resources using industry standards from NIST, Microsoft, CIS, DISA, etc.
2. **Create a strategy for systems hardening:** You do not need to harden all of your systems at once. Instead, create a strategy and plan based on risks identified within your technology ecosystem, and use a phased approach to remediate the biggest flaws.
3. **Patch vulnerabilities immediately**: Ensure that you have an automated and comprehensive vulnerability identification and patching system in place.
4. **Network hardening:** Ensure your firewall is properly configured and that all rules are regularly audited; secure remote access points and users; block any unused or unneeded open network ports; disable and remove unnecessary protocols and services; implement access lists; encrypt network traffic.
5. **Server hardening:** Put all servers in a secure datacenter; never test hardening on production servers; always harden servers before connecting them to the internet or external networks; avoid installing unnecessary software on a server; segregate servers appropriately; ensure superuser and administrative shares are properly set up, and that rights and access are limited in line with the principle of least privilege.
6. **Application hardening:** Remove any components or functions you do not need; restrict access to applications based on user roles and context (such as with application control); remove all sample files and default passwords. Application passwords should then be managed via an application password management/privileged password management solution, that enforces password best practices (password rotation, length, etc.). Hardening of applications should also entail inspecting integrations with other applications and systems, and removing, or reducing, unnecessary integration components and privileges.
7. **Database hardening:** Create admin restrictions, such as by controlling privileged access, on what users can do in a database; turn on node checking to verify applications and users; encrypt database information— both in transit and at rest; enforce secure passwords; introduce role-based access control (RBAC) privileges; remove unused accounts;
8. **Operating system hardening:** Apply OS updates, service packs, and patches automatically; remove unnecessary drivers, file sharing, libraries, software, services, and functionality; encrypt local storage; tighten registry and other systems permissions; log all activity, errors, and warnings; implement privileged user controls.
9. **Eliminate unnecessary accounts and privileges:** Enforce least privilege by removing unnecessary accounts (such as orphaned accounts and unused accounts) and privileges throughout your IT infrastructure

**Active Directory and Domain Service / DNS**

**directory -** is a hierarchical structure that stores information about objects on the network. A directory, in the most generic sense, is a comprehensive listing of objects. A phone book is a type of directory that stores information about people, businesses, and government organizations. Phone books typically record names, addresses, and phone numbers.

**Active Directory (AD) -** is a Microsoft technology used to manage computers and other devices on a network. It is a primary feature of Windows Server, an operating system that runs both local and Internet-based servers.

**directory service -** is a hierarchical arrangement of objects which are structured in a way that makes access easy. However, functioning as a locator service is not AD’s exclusive purpose. It also helps organizations have a central administration over all the activities carried out in their networks.

**What is Active Directory?**

* Active Directory is a directory service.
* The term directory service refers to two things — a directory where information about users and resources is stored and a service or services that let you access and manipulate those resources.
* Active Directory is a way to manage all elements of your network, including computers, groups, users, domains, security policies, and any type of user-defined objects
* Active Directory is built around Domain Name System (DNS) and lightweight directory access protocol (LDAP)
* Active Directory clients use DNS and LDAP to locate and access any type of resource on the network. Because these are platform-independent protocols

**The two most important are:**

1. Users should be able to access resources throughout the domain using a single logon.
2. Administrators should be able to centrally manage both users and resources.

**Active Directory provides several different services, which fall under the umbrella of “Active Directory Domain Services, ” or AD DS.**

**Domain Services** – Stores centralized data and manages communication between users and domains; includes login authentication and search functionality

**Certificate Services** – It generates, manages and shares certificates. A certificate uses encryption to enable a user to exchange information over the internet securely with a public key.

**Lightweight Directory Services** – Supports directory-enabled applications using the open (LDAP) protocol.

**Directory Federation Services** – Provides single-sign-on (SSO) to authenticate a user in multiple web applications in a single session.

**Rights Management** – It controls information rights and management. AD RMS encrypts content, such as email or Word documents, on a server to limit access.

**Domain Controllers**

* A server that is running AD DS is called a domain controller. Domain controllers host and replicate the directory service database inside the forest. The directory service also provides services for managing and authenticating resources in the forest.

**FUNDAMENTALS OF ACTIVE DIRECTORY**

1. If a client wants to access a service or a resource, it does so using the resource’s Active Directory name. To locate the resource, the client sends a standard DNS query to a dynamic DNS server by parsing the Active Directory name and sending the DNS part of the name as a query to the dynamic DNS server.
2. The dynamic DNS server provides the network address of the domain controller responsible for the name. This is similar to the way static DNS currently operates — it provides an IP address in response to a name query.
3. The client receives the domain controller’s address and uses it to make an LDAP query to the domain controller. The LDAP query finds the address of the system that has the resource or service that the client requires.
4. The domain controller responds with the requested information. The client accepts this information.
5. The client uses the protocols and standards that the resource or service requires and interacts with the server providing the resource.

**Benefits of Using Active Directory in an Enterprise Environment**

* More fine-grained administration is possible. Instead of having many administrators with sweeping and widespread rights over all directories (as in User Manager for Domains), you may have administrators who have a great deal of authority over a particular directory or group of directories but few, if any, rights over other directories. Rights can be granted down to the attribute (object property) level.
* By using the global catalog, you can query various attributes of objects. For example, a particular object — a user name — can be located by querying one of its attributes — say, last name.
* Global groups and local groups have gone the way of the dodo bird (in a pure 2000 Server environment). Instead, you can create nested groups that can contain many levels of users with various individual rights and privileges.
* Fault tolerance is greater. Each controller maintains a copy of the directory database and the replication topology is in a ring structure so that there are always two possible paths for replication.
* Active Directory’s Class Store and Group Policy Editor (GPE) let users access and download applications to which they are entitled, regardless which machine they are sitting at. Active Directory’s Microsoft Installer (MSI) lets developers package applications for use with Active Directory.
* A domain controller can be moved to another site or to another domain without having to reinstall 2000 Server.

**Logical Domain Structure of Active Directory**

* Active Directory is best understood from bottom to top. As an organization becomes larger and more complex, bottom-level units can be joined to make higher-level units.
* For example, domains can be joined in a hierarchical way to make domain trees, and domain trees can be joined with trusts to make domain forests.

**Common Terminologies and Active Directory Concepts:**

**Schema** - A set of rules, the schema, that defines the classes of objects and attributes contained in the directory, the constraints, and limits on instances of these objects, and the format of their names.

**Global catalog -** A global catalog that contains information about every object in the directory. This allows users and administrators to find directory information regardless of which domain in the directory contains the data.

**Sites -** in AD DS represent the physical structure, or topology, of your network. AD DS uses network topology information, which is stored in the directory as site, subnet, and site link objects, to build  
the most efficient replication topology.

**Lightweight Directory Access Protocol**- AD is based on the Lightweight Directory Access Protocol (LDAP). This protocol provides a common language for clients and servers to speak to one another.

**Simple Objects** - include computers, groups, users, security policies, and user-defined objects. Objects have attributes, some of which are mandatory and some of which are optional.

**Query and Index Mechanism** - This feature ensures users can locate each other in the Active Directory. A perfect example is when you start typing a user’s email address in the client’s recipient field and the possible matches are displayed.

**Lightweight Directory Access Protocol** - Commonly abbreviated as LDAP, this protocol enables the Active Directory to communicate with LADP enabled directory services in the network.

**Replication Service** - As the name suggests, *replication* ensures the Domain Controller is replicated onto another Domain Controller, thereby having the same schema and catalog.

**Organizational Units**

* Organizational units (OUs) are a new object typewithin 2000 Server’s Active Directory. They are designed to reduce the number of domains in an organization. Ous are often used to replace domains and subdomains on systems migrating to Active Directory.
* OUs can also be nested, so that a section within a department can have its own O

**Domains** - group network objects and OUs into a unit with a security boundary. By default, security policies and settings don’t flow from one domain to another.

**Domain Trees** - are a hierarchical way to group domains. All domains have, by default, a two-way trust arrangement with all other domains. The domains share a hierarchical naming structure with the child domain name appended to the parent name in the fully-qualified domain name.

**Domain Forests -** Domain forests are groups of domain trees. The domain trees within the domain forest don’t share a naming structure, but a two-way transitive trust is created among the root (top-level) domains in each domain tree. Because the domains within the domain trees are all joined with two-way trusts, in effect, resources become available to any user within the domain forest. Domain trees within the domain forest also share a global catalog.

**Multi-users Basics**

Let’s first understand what an operating system is. An operating system is a big piece of software that acts as an interface between the user and the computer hardware, which does multiple functions such as:

* Memory Management
* File Management
* Processor Management

**Difference Between Stand-Alone Operating Systems And Server Operating Systems**

* A stand-alone operating system is a complete operating system that works on a desktop computer, notebook computer, or mobile computing device. Example : DOS, Windows
* A server operating system is an operating system that is designed specifically to support a network.

**Now what is a Multi-User Operating System**

* A multi-user operating system is an operating system that allows multiple users to connect and operate a single operating system.
* The users interact with it through terminals or computers that gave them access to the system through a network or machines such as printers.
* The operating system should have to meet the requirements of all its users in a balanced way, so that if any problem would arise with one user, it does not affect any other user in the chain**.**

**EXAMPLES OF MULTI-USER OPERATING SYSTEM:**

**Multiple Virtual Storage** – An operating system from IBM that runs on the Mainframe. It is widely used in Enterprise computing which involves high intensity I/O. Example: Banking, Insurance, and Aviation business.

**Unix** – An open system Architecture that is highly reliable for small and medium scale business computing solutions. Since it’s based on Open system Architecture, Tech giants have their version of Unix such as Solaris, AIX, even Mac OS. Example: Hospitality industry, Healthcare

**TYPES OF MULTI-USER OPERATING SYSTEM**

**Distributed Operating System**

* Distributed Operating system also known as distributed computing is a collection of multiple components located on different computers that interact, co-ordinates, and emulates as a single coherent system to the enduser. End-user will communicate or operate them with the help of the Network.
* This system divides resources in a way that multiple requests can be handled and in turn each, individual request can be satisfied eventually.
* **Example:** Electronic banking, and Mobile Apps can be sited as the best example of Applications that can be hosted on Distributed OS. Users can conduct numerous transactions using a single network from the comfort of their place.

**Time sliced system**

* It is the system where each user task is allocated to a short span of CPU time. In other words, each task is assigned to a short period. These time slices appear too small to the eyes of the user. The decision to run the next piece of job is decided by an internal component called the ‘Scheduler’. This scheduler identifies and executes the run instruction or task that needs to be executed depending on the priority cycle.
* The users can take turns and thus the Operating System will handle user’s requests among the users who are connected. This feature is not available in the Single User Operating System, where the user and the machine come in contact directly.
* **Example:** Mainframe, a practical exam of the time-sliced system, where a user will be allocated a specific time slice to perform a complex task.

**Multiprocessor system**

* Multiprocessor system is the one, where the system uses more than one processor at the same time. Since all the processors would be working side by side, the work would be completed at a pace that would be much faster than the single-user operating system.
* **Example:** We can take Windows Operating System as a practical example of a multiprocessing system where word processor, spreadsheets, music player etc., everything can be opened at the same time without affecting the efficiency of any application that is being opened.

**FEATURES OF MULTI-USER OPERATING SYSTEM**

**Multiple processing:** As described earlier, we can execute multiple programs at the same time.

**Sharing of resources:** It is the feature that can be mapped to time slicing. Multiple peripherals such as printers, hard drives, etc. can be shared or we can even share different files or data

**Processing of data at the back end:** This feature allows data to be processed at the back end when they are not allowed to be processed at the front end. This allows other programs to interact with the processor in the front end simultaneously.

**Invisibility:** Many functions of the multi-user operating system is invisible to the users. This is because of the aspects such as the OS is instinctive or it happens at the lower end such as formatting of the disk and so on.

**COMPONENTS OF MULTIUSER OPERATING SYSTEM:**

**Processer:** The core of the computer called CPU (Central Processing Unit) is otherwise called the brain of the computer. In large machines, CPU would require more ICS. Whereas on smaller machines, CPU is mapped in a single chip popularly called as Microprocessor

**Memory:** The physical memory that is present inside the computer is where storage happens. It is otherwise called Random Access Memory (RAM). The system can correct data that is present in the main memory, therefore every program that is executed must be copied from a physical storage such as hard disk. Main memory is always marked important because, it determines how many programs can be executed at one time and the amount of data that would be available.

* **Hard Disks**: It can hold large amount of data and how also determines how many programs can be run at a single time.
* **Floppy Disks**: It is inexpensive but can hold only less data when compared to hard drives. It is also portable.
* **Optical Disks:** They use Lasers to read and write data. They can hold large data like hard disks but they are not portable like hard drives. CD (Compact Disk) is used to write and read files.
* **Tapes:** They are inexpensive as well. They hold large memory but data access in random cannot be done.

**Terminals:**

**Dumb Terminal:**

* It is featured with its computer and keyboard.
* It does not have the processing power.
* They are used to do remote work on mainframe systems.

**Smart Terminals:**

* Simple editing and processing can be done using a smart terminal. They are inexpensive but do not have any physical storage devices such as hard drives.

**Features and Characteristics of Multi User Operating System**

* **Multi Tasking –** Multi user O/S is capable to perform couple of tasks at concurrently, and multiple programs can be run on this operating system at a same time.
* **Resource Sharing** – In the multiuser operating system, several peripheral can be shared such as printers, fax m/c, plotters, and hard drives etc. Due to this feature, users can share own documents. In this system, tiny time slice of CPU time is allotted to all users.
* **Background Processing** – In which, if given instructions are not processed, and then they perform their tasks in the background as well as other programs are interacting with system in the current time

**Advantages of Multi User OS**

* Multi user operating system is very useful in offices or library, because it can be handled printing jobs with effective manner.
* Multiple users can access same copy of document on one computer system. For example, if some PPT file is stored in the one computer, then other user can watch this PPT on other terminals.
* Multi user O/S is used in the ticket reservation system.
* Multi user O/S is also used in the some functionality of Airlines.
* Entire system does not get halt, if one computer gets fails in own network system.
* All company’s managers also can use the multi user operating system for searching manufacturing records, because this record is stored in one system but staff no bound on one computer, they can access other terminals those records.

**Disadvantage of Multi User OS**

* If, virus attacks on one computer, then this virus spreads on entire network system simultaneously, and finally all computer system can get fails.
* All information of computer is shared publicly, so your private data is shared on the entire network.

**Configuration Management**

**Configuration Management is a set of interrelated processes, management techniques, and supporting tools that assure:**

1. our work products (what a company is managing) are as they should be and conform to requirements;
2. changes to the above are properly evaluated, authorized, and implemented;
3. all information / data necessary to manage our end items and other related work products is:kept current and accurate,  
    properly structured for users’ needs, and  
    readily available to all who need to know.

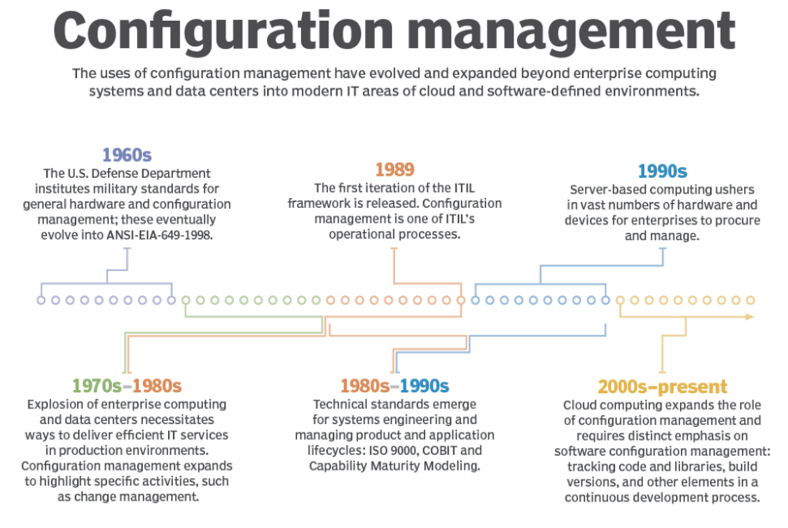
**Configuration Management (CM)** ensures that an organization is making informed business decisions, performing correct actions, and that all work products are what they are intended to be at every point in the lifecycle.

CM is about knowing what we did yesterday, what we are doing today, and what we will be doing in the future, as well as understanding the reasoning and authority behind every action/change that got us there.

**How does configuration management work?**

* For a configuration management system to operate, it needs some form of mechanism in which to store the information it governs.
* Originally, this was called the*configuration management database*(CMDB); ITIL v3 introduced the concept of a configuration management system (CMS) to replace the CMDB.
* The CMDB promotes the concept of a singular unchanging or immovable repository, while the CMS provides a conceptualized system of CMDBs that act together to support the needs of this governance process.
* Both demonstrate advantages over a static CM spreadsheet or a text file that requires significant manual upkeep and cannot integrate base workflows and best practices.
* Every service management tool is deployed with a supporting data repository.
* Without the governance process of configuration management validating its contents, the repository is simply an operational database with unverified data, not a CMDB or CMS.
* Automated configuration audit and verification components entitle a repository to be leveraged as an authorized gold source of assets. A manual audit is also possible.
* Every service management tool is deployed with a supporting data repository.
* Without the governance process of configuration management validating its contents, the repository is simply an operational database with unverified data, not a CMDB or CMS.
* Automated configuration audit and verification components entitle a repository to be leveraged as an authorized gold source of assets. A manual audit is also possible.

**History of Configuration Management**

****

**Why do we use it?**

* Source of Track
* Track configurations in a Version Control System
* Reproducible Configurations and Infrastructure Deployment

**Configuration Management Solution  
(Automation Concept)**

**METHODS OF CONFIGURATION MANAGEMENT:**

1. Resources and destroyed and recreated (immutable)
2. Changes are made idempotently over a resource and converge

**IDEMPOTENCY**

* Automated configuration scripts produce exact result every time
* Configuration changes occur “in place”
* Only if a resource does not match requirement is it changed
* There is no fear of breaking existing resources
* Ansible and Chef are idempotent CMSs

**CONVERGE**

* Ability to successfully modify or restore when necessary
* When resources diverge overtime converge brings these resources back to specification
* A node is said to have “converged” when resources are changed in this manner

**IMMUTABILITY**

* The concept that resources are to be created and destroyed : never changed
* Immutable infrastructure is similar to blue/green deployment concept
* Terraform and CloudFormation work in this manner

**What are the risks of not using configuration management?**

* Imagine an IT environment without configuration management:
* A data center operates workloads, but IT administrators and business leaders have no single source of truth about the hardware and software makeup across the enterprise.
* Staff know some of the hardware and software setups but must directly inspect each data center element to determine the existing configuration.
* Imagine an IT environment without configuration management:
* A data center operates workloads, but IT administrators and business leaders have no single source of truth about the hardware and software makeup across the enterprise.
* Staff know some of the hardware and software setups but must directly inspect each data center element to determine the existing configuration.

**REMOTE CONFIGURATION**

**Remote Assistance vs. Remote Desktop: Understanding the Difference**

**Remote Assistance**

* Remote assistance refers to a connection that is intended to provide technical support from a distance.
* A user who is sitting at his or her computer can invite a technician to see what is happening on the screen remotely. The remote user receives an invitation from the host and cannot log in without responding to it.

**Remote Desktop**

* Remote desktop refers to the connection made when a user obtains full access to a host computer or device from a distance. For example, a user connects to their work computer from a personal computer at home.
* In a typical windows remote desktop connection, the host computer’s screen locks when the session begins. Only the remote computer will see any video output.
* If someone makes a remote connection and saves a file to a hard drive, it will be on the host computer and would have to be transferred to the remote computer when the connection has ended.
* aims to increase efficiency whereas remote assistance facilitates technical support.

**How to Connect Remote Assistance**

1. The host user sends an invitation to the technician.
2. The technician, using the remote computer, must accept this invitation in order to gain access to the host computer.
3. Depending on the configuration, the host user may choose to simply share their screen or give full control over to the technician.
4. The connection continues until either side ends the session.

**A Windows remote desktop connection involves a single user operating another computer at a distance, which means the connection process is going to look a little different:**

* Prior to leaving the host computer, the user must navigate to the control panel and toggle the option to allow remote access.
* Once the host computer is configured to allow remote access, the user must have proper credentials at the ready.
* From the remote computer, the user is prompted to enter the IP address of the host.
* Then, the user is prompted to enter the login credentials of the host.
* At this point, the remote screen displays information from the host computer, allowing the user to interact with it like normal, even if the computer is miles away.

**Security Concerns of Remote Assistance**

* The primary difference is the nature of the connection. Unless the host user is acting improperly, there is always someone on either side of the connection. The host user must invite the connection and monitor it throughout the session.
* when someone is giving remote desktop assistance and wants to take control of functions like the keyboard and mouse, it's important to make sure the host user actively grants those permissions.

**Security Concerns of Remote Desktop**

* It's important to make sure remote workers use a VPN or something similar to keep connections encrypted–and monitor the network for unusual activity. If the remote user's device is compromised, it could allow a malicious actor easy access to the corporate network.
* If a cybercriminal can obtain the IP address of a computer, that person then only needs to obtain the login credentials. If an employee uses a weak password, a criminal can log in and potentially access information and data on the work computer–or elsewhere on the network.
* This is why many IT teams implement strong authentication protocols, such as multi-factor authentication, to make sure any users who try to remotely connect are actually who they say they are.

**Securing Remote Desktop (RDP) for System Administrators  
Basic Security Tips for Remote Desktop:**

1. Use strong passwords
2. Use Two-factor authentication
3. Update your software
4. Restrict access using firewalls
5. Enable Network Level Authentication
6. Limit users who can log in using Remote Desktop
7. Set an account lockout policy