**System administration is about putting together a network of computers (workstations, PCs and supercomputers), getting them running and then *keeping them running in spite of the* activities of *users who tend to cause the systems to fail.***

System administration is a service profession, but it is far more than that. System administrators are also mechanics, sociologists and research scientists.

**Challenges of network administrator**

System administration is not just about installing operating systems. It is about planning and designing an efficient *community of computers so that real users will be able to get their* jobs done.

Designing a network which is logical and efficient.

• Deploying large numbers of machines which can be easily upgraded later.

• Deciding what services are needed.

• Planning and implementing adequate security.

• Providing a comfortable environment for users.

• Developing ways of fixing errors and problems which occur

**Bugs**

Bugs can be caused by many things. They may come from

• Shoddy software.

• Little known problems in the operating system.

• Unfortunate clashes between incompatible software, i.e. one software package destroys

the operation of another.

• Totally unexplainable phenomena, cosmic rays and invasions by digital life-forms.

**Information Sources to Sysadmins**

Information can be found from many sources:

• Printed manuals.

Unix manual pages (man and apropos commands).

• The World Wide Web.

• RFCs (Requests for comment), available on the web.

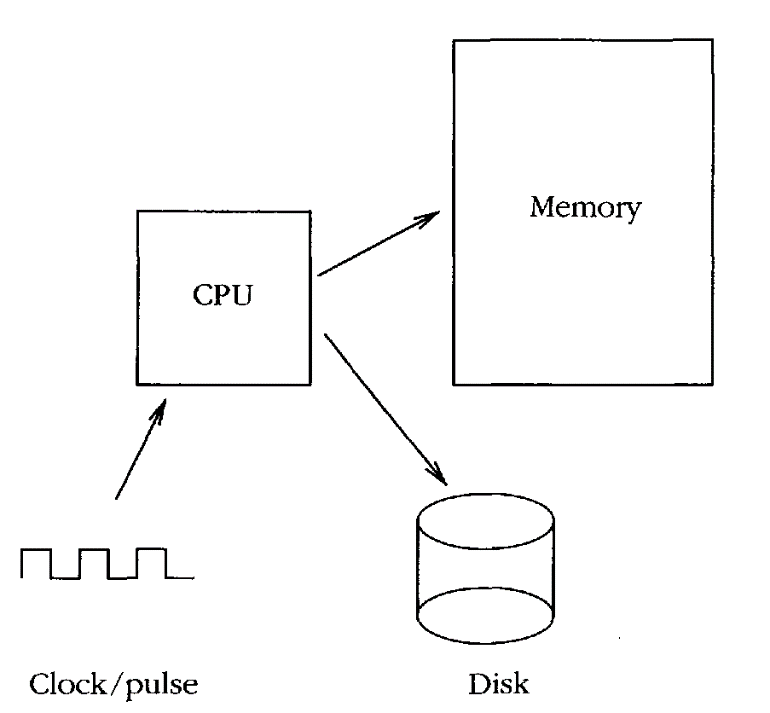
• News groups and discussions.

• Papers from the SAGE/Usenix LISA conferences [16].

• More specialized books.

**System Components**

**System a lot to refer both to the operating system of a computer and often, collectively the set of computers on a network.**

**Von Neumann Architecture**

**Each computer has a clock which drives a *central processor unit*(CPU), *random access memory (RAM) and an array of other devices, such as disk drives***

In order to make these parts work together, the CPU is designed to run programs which

can read and write to hardware devices.

The most important program is the **operating system kernel.**

An operating system is the software which shares and controls the hardware resources of a computer.

It shields the user of the machine from the low-level details of the machine's

operation and provides frequently needed facilities

Normally the operating system has a number of key elements: (i) a technical layer of software for driving the hardware of the computer, like disk drives, the keyboard and the screen; (ii) a filesystem which provides a way of organizing files logically; and (iii) a simple user interface which enables users to run

their own programs and to manipulate their files in a simple way.

Of central importance to an operating system is a core software system or *kernel which is* responsible for allocating and sharing the resources of the system between several running programs or *processes.*

It is supplemented by a number of supporting *services (paging, RPC,* FTP, WWW, etc.) which either assist the kernel or extend its resource sharing to the network domain.

In [computer](http://en.wikipedia.org/wiki/Computer) [operating systems](http://en.wikipedia.org/wiki/Operating_system), **paging** is one of the [memory-management](http://en.wikipedia.org/wiki/Memory_management) schemes by which a computer can store and retrieve data from [secondary storage](http://en.wikipedia.org/wiki/Computer_data_storage) for use in [main memory](http://en.wikipedia.org/wiki/Computer_data_storage). In the paging memory-management scheme, the operating system retrieves data from secondary storage in same-size [blocks](http://en.wikipedia.org/wiki/Block_(data_storage)) called *pages*

A **distributed operating system** is the logical aggregation of [operating system](http://en.wikipedia.org/wiki/Operating_system) software over a collection of independent, [networked](http://en.wikipedia.org/wiki/Computer_network), [communicating](http://en.wikipedia.org/wiki/Inter-process_communication), and physically separate computational nodes.

**Secure os**

The most fundamental tenet of security is the ability to restrict access to certain system resources. The main reason why DOS, Windows 9x and the Macintosh are so susceptible to virus attacks is because any user can change the operating system's files

To restrict access to the system we require a notion of *ownership and permission. Ordinary* users should not have access to the hardware devices of a secure operating system's files, only their own files, for then they will not be able do anything to compromise the security of the system.

Secure operating systems are usually multi-user systems, i.e. operating systems where files and processes can be owned by a particular user, and access is restricted on the basis of user identity.

**Unix File Model**

**Unix has a hierarchical file system which makes use of directories and sub-directories to form a tree.**

**All file systems on Unix-like operating systems are based on a system of *index nodes,* or *inodes, in which every file has an index entry stored in a special part of the file system***

**The top or start of the Unix file tree is called the root file system or '/'.**

**The File Hierarchy**

**The main sub-directories of the root directory together with the most important file are shown below**

• **/b** in Executable (binary) programs. On most systems this is a separate directory to /usr/bin. In SunOS, this is a pointer (link) to /usr/bin.

• **/etc** Miscellaneous programs and configuration files. This directory has become very messy over the history of Unix and has become a dumping ground for almost anything. Recent versions of unix have begun to tidy up this directory by creating subdirectories /etc/mail, /etc/services, etc!

• **/usr** This contains the main meat of Unix. This is where application software lives, together with all of the basic libraries used by the OS.

• **/usr/bin** More executables from the OS.

• **/usr/local** This is where users" custom software is normally added.

**/sbin** A special area for statically linked system binaries. They are placed here to

distinguish commands used solely by the system administrator from user commands, and so that they lie on the system root partition where they are guaranteed to be accessible during booting.

• **/sys** This holds the configuration data which go to build the system kernel

• **/export** Network servers only use this. This contains the disk space set aside for client machines which do not have their own disks. It is like a 'virtual disk' for diskless clients.

• **/dev and /devices** A place where all the 'logical devices' are collected. These are

called 'device nodes' in Unix and are created by mknod. Logical devices are Unix's official entry points for writing to devices. For instance, /dev/console is a route to the system console, while /dev/kmem is a route for reading kernel memory. Device nodes enable devices to be treated as though they were files.

• **/home** (called /users on some systems.) Each user has a separate login directory

where files can be kept. These are normally stored under /home by some convention decided by the system administrator.

**• /root** On newer Unix-like systems, root has been given a home-directory which is no longer the root of the file system '/'. The name root then loses its logic.

Every unix directory contains two 'virtual' directories marked by a single dot and two dots:

**Is *–a***

***. ..***

The single dot represents the directory one is already in (the current directory). The double dots mean the directory one level up the tree from the current location

**Symbolic Links**

A symbolic link is a pointer or an alias to another file. The command

**In -s fromf ile /other/directory/tolink**

makes the file fromfile appear to exist at /other/directory/tolink simultaneously.

**File Access Control**

To restrict privilege to files on the system, and create the illusion of a virtual host for every logged-on user, Unix records information about *who creates files and also who is allowed to* access them later.

A file's contents are classified by *magic numbers which are codes kept in the file's* inode and defined in the magic number file for the system

Each user has a unique *usemameor loginname, together with a unique user id or uid.*. A file belongs to user A if it is *owned by user A. User A then decides*

whether or not other users can read, write or execute the file by setting the *protection bits or* the *permission of the file using the command* ***chmod.***

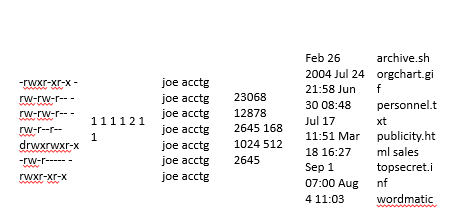
**$ chmod [*options*] *mode*[,*mode*] *file1* [*file2* ...]**

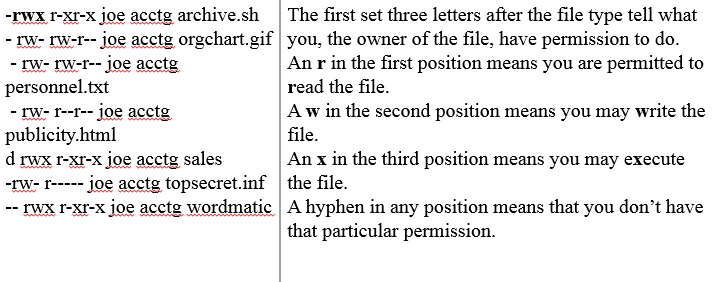
In addition to user identities, there are groups of users. The idea of a group is that several named users might want to be able to read and work on a file, without other users being able to access it

**Changing File Permissions**

Let’s take a closer look at the contents of a sample directory by typing the command

[ls -l](http://catcode.com/teachmod/placeholder.html?the%20ls%20command) (the “l” stands for “long”.)





Every user is a member of at least one group, called the *login group, and each*

group has both a textual name and a number *(group id). The uid and gid of each user is sn*recorded in the file /etc/passwd

**Making Programs Executable**

A Unix program is normally executed by typing its pathname. If the x execute bit is not set on the file, this will generate a 'Permission denied' error

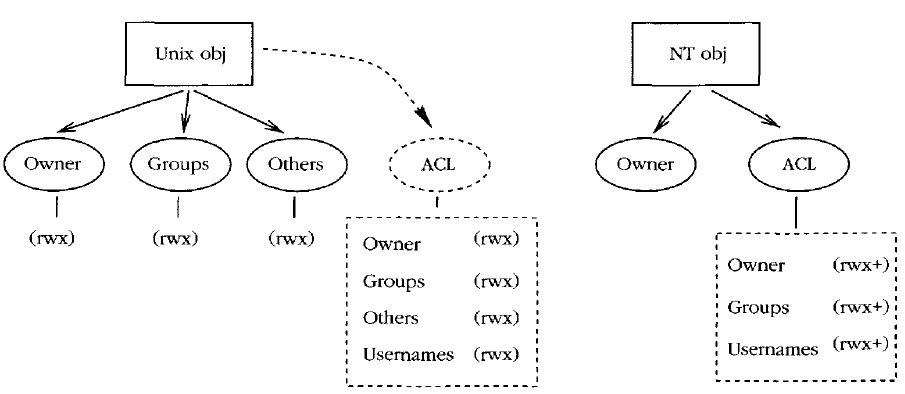
**chmod u+x *filename***

This command would set execute permissions for the owner of the file;

**chmod ug+x *filename***

would set execute permissions for the owner and for any users in the same group as the file.

**Access Control Lists**



**NT File Model**

The file system NTFS was introduced with NT in restricting access to files.

NTFS, like the Unix file system, is a hierarchical file system with files and directories. Each file or directory has an owner, but no group membership

**File System Layout**

Drawing on its DOS legacy, NT treats different disk partitions as independent disks, labeled by a letter of the alphabet:

C: D: ...

C : is the primary hard disk partition.

The system root is usually stored in

C : \WinNT, and is generally referred to by the system environment variable %System-Root%:

• **C : \I386** This directory contains binary code and data for the NT operating system. This should normally be left alone.

• **C:\Progr am Files** This is NT's official location for new software. Program packages which you buy should install themselves in subdirectories of this directory. More often than not, they choose their own locations, often with a distressing lack of discipline.

• **C : \Temp** Temporary scratch space, like Unix's /tmp.

• **C: \WinNT** This is the root directory for the NT system. This is mainly for operating system files, so you should not place new files under this directory yourself unless you really know what you are doing. Some software packages might install themselves here.

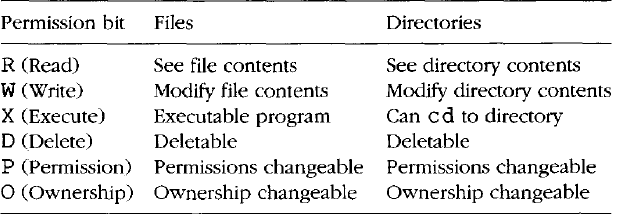
• **C:\WinNT\config** Configuration information for programs. These are generally

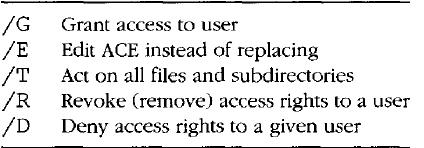
binary files, so the contents of NT configuration files is not very interesting.

• **C: \WinNT\system32** This is the so-called system root. This is where most system applications and data files are kept.

**Access Control Lists**

NT files and directories have the following attributes. Access control lists are composed of Access Control Entries (ACEs), which consist of these:





**Logs and Audits**

Operating system kernels share resources and offer services. They can be asked to keep lists of transactions which have taken place so that one can later go back and see exactly what happened at a given time. This is called logging or auditing.

The use for auditing is so-called *non-repudiation, or non-denial.* If everything on a system is logged, then users cannot back away and claim that they did not do something: it's all there in the log. Non-repudiation is a security feature which encourages users to be responsible for their actions.

**Privileged Accounts**

Operating systems which restrict user privileges need an account which can be used to configure and maintain the system. Such an account must have access to the whole system, without regard for restrictions. It is therefore called a privileged account.

**Principle 1 (Privilege) *Restriction of unnecessary privilege protects a system from accidental*** *and malicious damage, and infection by viruses, and prevents users from concealing their actions with false identities. It is desirable to restrict users' privileges for the greater good of everyone on the network.*

**Corollary 2 (Privilege) *No one should use a privileged root/Administrator account as a user*** *account. To do so is to place the system in jeopardy*

One of the major threats to Internet security has been the fact that everyone can now be root/Administrator on their own host. Many security mechanisms associated with trusted ports, TCP/IP spoofing, etc., are now broken, since all of the security of these systems lies in the outdated assumption that ordinary users will not have privileged access to network hardware and the kernel.

**Hardware Awareness**

To be a system administrator it is not absolutely essential to know much about hardware, but it is very useful to have a basic appreciation of hardware installation procedures and how to nurse-maid hardware later

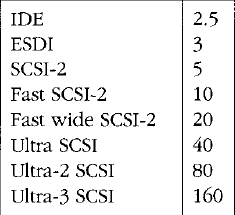
• *Read instructions: when dealing with hardware, one should always look for and read* instructions in a manual.

• *Interfaces and connectors: hardware is often connected to an interface by a cable or* connector. Obtaining the correct cable is of vital importance. Many manufacturers use cables which look similar, superficially, but which actually are different. An incorrect cable can result in damage to an interface.

Network interfaces are often built-in; they can always be added with expansion cards.

• *Handling components: modern day CMOS chips work at low voltages (typically 5 volts* or lower). Standing on the floor with insulating shoes, you can pick up a static electric charge of several thousand volts. Such a charge can instantly destroy computer chips.

*Disks: the most common disk types are IDE (integrated drive electronics) and SCSI (small* computer software interface). IDE disks are usually cheaper than SCSI disks, but SCSI disks are more efficient at handling multiple accesses, and are therefore better in multitasking systems.



• ***Memory****: memory chips are sold on small circuit boards called SIMMs. These SIMMs are* sold in different sizes, and with different speeds. When buying and installing RAM, remember that

- The physical size of SIMMs is important. Most have 72 pins and some older SIMMs have 30 pins.

- SIMMs are sold in 1MB, 4MB, 16MB, 64MB sizes, etc. Find out what size you can use in your system. In most cases you are not allowed to mix different sizes.

- Do not buy slower RAM than that which is recommended for your computer, or it will not work.

- There are several incompatible kinds of RAM, FP RAM, EDO RAM, SDRAM, such as which work in different ways. ECC/SDRAM RAM (error correcting code, synchronous dynamic RAM) is tolerant to error from external noise sources like cosmic rays, etc. It can be recommended for important servers.

— On some computers you need to fill up RAM slots in a particular order, otherwise the system will not be able to find them.

**System Uniformity**

**Principle 3 (Uniformity) *A uniform configuration minimizes the number of differences and*** *exceptions one has to take into account later. This applies to hardware and software alike*

PC networks are often a melange of random parts from different manufacturers. If possible, one should standardize graphics and network interfaces, disk sizes, mice and any other devices which have to be configured.

This means that, not only will it be easier to configure and maintain, but also that it 'will be easier to buy extra parts or cannibalize systems for parts

later.

With software, the same principle applies: a uniform software base is easier to install and maintain than one in which special software needs to be configured in special ways.

**Networked Communities**

**Communities**

System administration is not just about machines and individuals, it is about communities. There is the local community of users on multi-user machines; then there is the local area network community of machines at a site; finally, there is the global community of all machines and networks in the world.

**Principle (Communities) *What one member of a cooperative community does affects*** *every other member, and vice versa. Each member of the community therefore has a responsibility to consider the well-being of the other members of the community.*

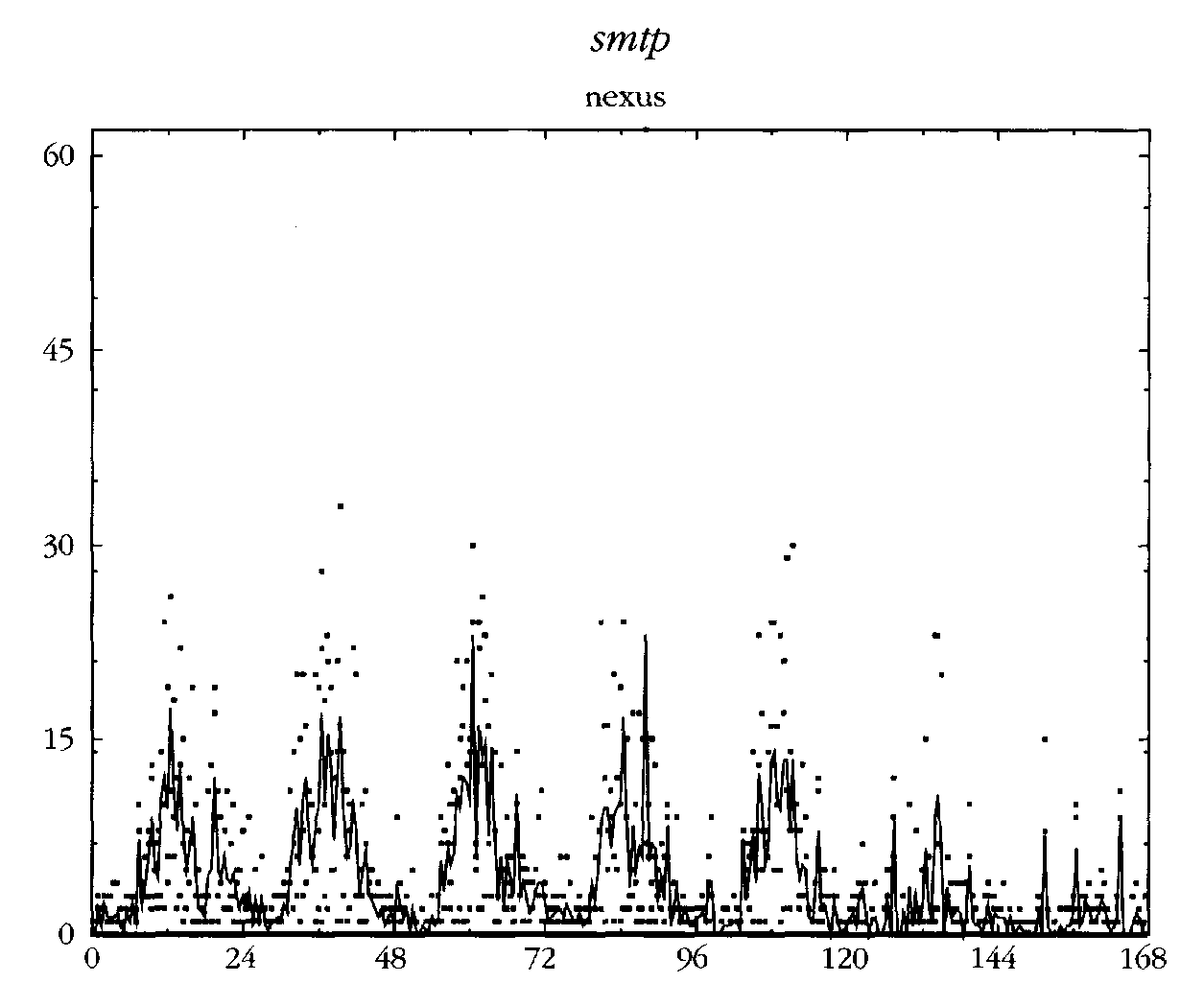
**Principle (Multi-user communities)** *A multi-user computer system does not belong to any one user. All users must share the resources of the system. Each user has a responsibility to consider the effect of his/her actions on all the other users.*

**Principle 6 (Network communities) *A computer which is plugged into the network is no*** *longer just ours. It is part of a society of machines which shares resources and communicates with the whole. What that machine does affects other machines. What other machines do affects that machine.*

**User Sociology**

A newly installed machine does not usually require attention until it is first used, but as soon as a user starts running programs and storing data, the reliability and efficiency of the system are tested. This is where the challenge of system administration lies.

The load on computers and on networks is a social phenomenon: it peaks in response to patterns of human behavior. For example, at universities and colleges network traffic usually peaks during lunch breaks, when students rush to the terminal rooms to surf on the web or to read e-mail.



**Client-Server Cooperation**

At the heart of all cooperation in a community is a system of *centralization and delegation.* No program or entity can do everything alone, nor is it expected to do so. It makes sense for certain groups to specialize in performing certain jobs. That is the function of a society

**Principle (Delegation I) *Leave experts to do their jobs. Assigning responsibility for a task to*** *a body which specializes in that task is an efficient use of resources*

The client-server nomenclature has been confused by history. A server is not a host, but a program or process which runs on a host. A client is any process which requires the services of a server.

**Host Identities and Name Services**

A host can have all of the following:

• *Host ID: circuit board identity number. Often used in software licensing.*

• *Install name: configured at install time. This is often compiled into the kernel, or placed* in a file like /etc/hostname. Solaris adds to the confusion by also maintaining the install name in /etc/hostname. le0 or an equivalent file for the appropriate network interface, together with several files in /etc/net/\*/hosts.

• *Application level name: any name used by application software when talking to other* hosts.

• *Local file mapping: originally the Unix /etc/hosts file was used to map IP addresses* to names, and *vice versa. Other systems have similar local files, to avoid looking up on* network services.

• *Network Information Service: a local area network database service developed by Oracle*. This was originally called Yellow Pages, and many of its components still bear the 'yp' prefix.

• *Network level address(es): each network interface (Ethernet/FDDI, etc.) has a hardware* address burned into it at the factory, also called its MAC address, or Media Access Control address. Some services (e.g. RARP) will turn this into a name or an IP address through a secondary naming service like DNS.

• *DNS name(s): the name returned by a domain name server (DNS/BIND) based on an IP* address key.

• ***WINS*** *name(s): the name returned by a WINS server (Microsoft's name server) based on* the IP address.

At boot-time, then, each host needs to obtain an Internet identity. It has two choices:

• Ask for an address to be provided from a list of free addresses (DHCP or **BOOTP** protocols).

• Always use the same IP address, stored on its system configuration files (requires correct information on the disk).

**Windows Internet Name Service** (WINS) is [Microsoft](https://en.wikipedia.org/wiki/Microsoft)'s implementation of [NetBIOS Name Service](https://en.wikipedia.org/wiki/NBNS) (NBNS), a name server and service for [NetBIOS](https://en.wikipedia.org/wiki/NetBIOS) computer names. Effectively, WINS is to NetBIOS names what [DNS](https://en.wikipedia.org/wiki/Domain_Name_System) is to [domain names](https://en.wikipedia.org/wiki/Domain_name) — a central mapping of host names to network addresses. Like the DNS, it is implemented in two parts, a server service (that manages the [embedded Jet Database](https://en.wikipedia.org/wiki/Microsoft_Jet_Database_Engine), server to server replication, service requests, and conflicts) and a [TCP/IP](https://en.wikipedia.org/wiki/TCP/IP) client component which manages the client's registration and renewal of names, and takes care of queries.

The Bootstrap Protocol (**BOOTP**) is a computer networking protocol used in Internet Protocol networks to automatically assign an IP address to network devices from a configuration server.

The only worldwide service in common use today is DNS (the Domain Name Service) whose common implementation is called BIND (Berkeley Internet Name Domain). This associates IP addresses with a list of names. Every host in the DNS has a *canonical name,* or official name, and any number of *aliases. For instance, a host which runs several important* services might have the canonical name.

DNS binds a local network to the worldwide Internet in several important ways. It makes it possible for data to organizations to be spread across the surface of the planet at any location, and yet still maintain a transparent naming structure

Under NT, each system has an alphanumeric name which is chosen during the installation. A domain server will provide an SID (Security ID) for the name which helps prevent spoofing.

When NT boots it broadcasts the name across the network to see whether it is

already in use. If the name is in use, the user of the workstation is prompted for a new name(!)