An Overview of the Novell NetWare Operating System and Others Existing Networking Operating System

Aditi Chakraborty
Lecturer, Panskura Banamali College,
Panskura,Paschim Medinipur,W.B. INDIA.
aditichch@gmail.com

Abstract.

The Novell NetWare [1] operating system is designed specifically to provide service to clients over a computer network. This design has resulted in a system that differs in several respects from more general-purpose operating systems and also from the others existing networking operating system, such as Windows NT Server, UNIX/LINUX, and MAC OS Apple Share. In addition to highlighting the design decisions that have led to these differences, this paper provides an overview of the NetWare operating system, with a detailed description of its kernel and its software-based approach to fault tolerance.

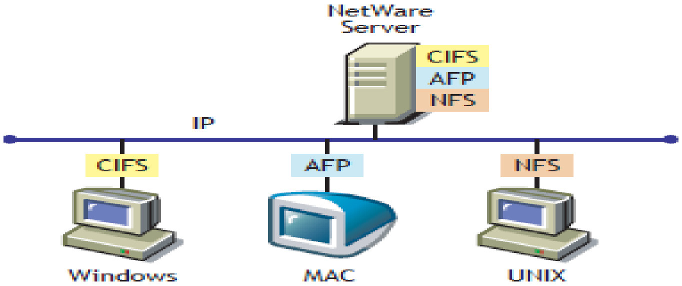
Keywords: Operating system, network, server, Windows, Novell, UNIX/LINUX, and Mac.

1. Introduction

Computers have operating system software [2] that allows them to function. Without operating system instructions, a computer is nothing more than a box with circuits. This is the same with networks. Without a Network Operating System, a network is nothing more than a number of computer devices connected together. In order to transmit information and communicate across a network, it is necessary to have a Network Operating System. There are several different Network Operating Systems, each with its own set of features and protocols.

The NetWare operating system (NetWare OS) [1] was originally designed in 1982-83 and has had a number of major changes over the intervening ten years, including converting the system from a Motorola 68000-based system to one based on the Intel 80x86 architecture. Novell Native File Access Protocols lets Macintosh, Windows, and UNIX workstations access and store files on NetWare servers without having to install any additional software—such as the Novell ClientTM. The software is installed only on the NetWare server and provides "out of the box" network access. Just connect the network cable, start the computer, and you have access to servers on your network. No client software installation. No client configuration. No problem.

Novell Native File Access Pack software enables the NetWare server to use the same protocol (referred to as *native*) as the client workstation to copy, delete, move, save, and open files. Windows workstations perform these tasks using the native Common Internet File System (CIFS) protocol, Macintosh workstations use the native Apple* Filing Protocol (AFP), and UNIX computers use the Network File System (NFS) protocol.



Enabling native protocols on a NetWare server means that users can access files on the network, map network drives, and create shortcuts to NetWare servers using the native methods available in their specific operating system. Windows users can use their familiar Network Neighborhood (or My Network Places). Macintosh users can use Chooser or the Go menu to access network files and even create aliases. Because the NetWare server is running native protocols, users can copy, delete, move, save, and open network files—just like they would if they were working locally.

2. Novell NetWare system

A NetWare system [1] [5] (also known as a “NetWare network” or “NetWare LAN”) provides for the sharing of services over a network. The two principle components of the system are clients, which request service, and servers, which arbitrate the requests and provide the service.1 Typical services include file systems, files, printer queues, printers, mail systems, and databases. The network can either be a single subnet, or a collection of subnets interconnected by routers.

In a NetWare system, clients run “off the shelf” operating systems such as DOS, Windows, OS/2, UNIX, and the Macintosh operating system.

A NetWare server consists of the NetWare OS kernel and a number of NetWare Loadable Modules (NLMs). An NLM is similar to an “a.out” file in the Unix operating system [3]: it consists of code, data, relocation, and symbol information for a program. An NLM is generated by a linkage editor, which uses as input one or more object files as well as a control file. The object files are generated by assemblers and/or compilers from source files. The relocation information allows an NLM to be loaded at any address in the system.

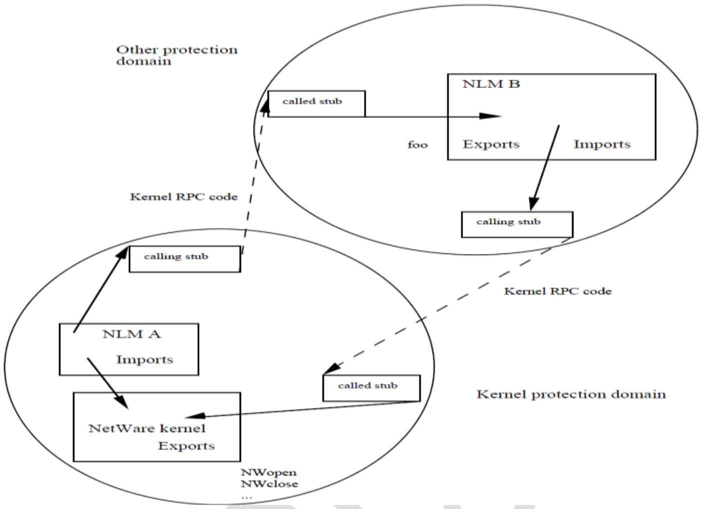
The NetWare OS does not have the concept of a process. Instead, the basic units are the NLM, a protection domain, and a thread.

2.1. The NetWare Kernel

The NetWare kernel [4] includes a scheduler, a memory manager, and a primitive file system with which to access other pieces of the system during the process of booting up the system, and a loader that is used to load NLMs. In contemporary usage, the NetWare kernel might be known as a microkernel [4].

2.2. The Loader

Currently, all NLMs share the same address space, though this address space can be divided up into various protection domains. By default, all NLMs are loaded into and run in the kernel domain. At the time an NLM is loaded, however, the user can specify a protection domain into which to load the NLM. All NLMs resident in a given protection domain have shared memory access to each other's memory. The kernel domain has read/write access to memory in all other protection domains.



The loader maintains a symbol table in order to resolve external references when loading NLMs. When the system first loads, this symbol table contains one entry for each procedure exported by the NetWare kernel.

When an NLM contains a call to a procedure in an NLM that is loaded in a different protection domain, an intra- machine remote procedure call (RPC) [BIRR84] is made to invoke the procedure.

2.3. The NLM Execution Environment

The NLM Execution Environment provided by the Kernel When first loaded, an NLM is given one thread of control. It can then create as many additional threads as it needs.

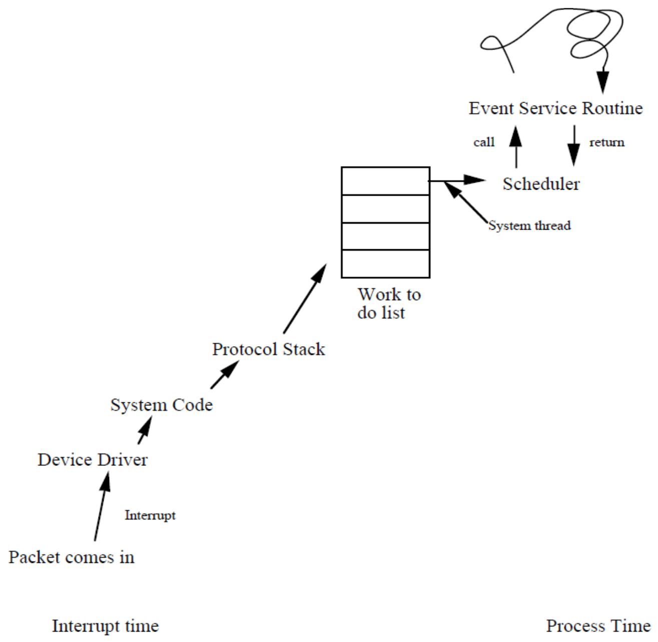
2.4. The Scheduler

The NetWare OS was designed specifically for the purpose of providing service to networked clients. The performance of servicing network requests is very sensitive to the scheduling policy of the server operating system.

The NetWare scheduler is therefore tuned to give high performance in this specific environment.

As mentioned above, the NetWare scheduler provides a non-preemptive scheduling paradigm for the execution of threads. A thread that has control of the machine will not have this control taken away from it unless one of the following occurs:

- it has called one of the yield calls (see below)
- it has blocked (awaiting disk I/O, for example)
- it has returned to the scheduler (for threads dispatched to service work objects)



In particular, interrupts (including timer interrupts) do not cause a thread switch until, at least, the next yield call.

3. Windows for Workgroups/Windows 95/Windows NT Server

3.1. Windows for Workgroups/Windows 95

Windows for Workgroups, introduced in the early 90s and Windows 95, introduced in 1995 are considered peer-to-peer networking systems and do not have the capabilities of true internetworking operating systems. They are, however, inexpensive and more than adequate for small workgroups wanting to share resources, use email, and connect to the Internet.

Windows for Workgroups and Windows 95 both offer peer-to-peer network protocols (Windows NT will be discussed later). The protocols used by these operating systems allow users to share files and devices over LANs.

Both offer NetBEUI, Microsoft's small network protocol. They also offer TCP/IP and IPX/SPX protocols to access the network through either a dialup connection/modem, or directly through a NIC. NetBEUI protocols, while not routable, are more than adequate to meet small LAN needs. NetBEUI protocols are easy to use and do not require in-depth networking knowledge. NetBEUI software identifies computer devices by name and it is certainly easier to remember that a computer's name is Juanita or Justin than 141.252.20.2 or 141.252.10.1. Each device name in a network must be unique. NetBEUI software installed on each of the networked computers is all that is necessary to configure devices in order to share resources and create a network. If a small company does want to connect to the Internet, the necessary software and protocols are available with these operating systems.

Shared resources on Windows for Workgroups/95 networks are accessed by a password that protects the resource and there is only one level of access;

either you have access or you don't have access. Also user-by-user passwords are not part of the protocols unless Windows NT is present. What this means is that anyone connected to the network who knows the password of the resource has access to that resource. This can create security issues since there is no way to prevent a user from access once s/he knows the password. As the network grows, it is usually more difficult to keep resource passwords secure. Since there is no central control, managing these peer-to-peer networks becomes an issue when the network becomes too large.

3.2. Windows NT Server

Windows NT Server was first introduced in the mid-90s and is capable of managing workgroups just like Windows for Workgroups/95. It is a client/server networking operating system that uses routable protocols, making it a true Internetworking Operating System enabling network administrators to connect LANs to WANs. Windows NT Server also provides services for OS/2 and Novell NetWare clients and is able to run on various microprocessors such as Intel Pentium, DEC Alpha, RISC, MIPS, and PowerPC based systems.

Windows NT Server has all of the advantages mentioned for the other Windows operating systems, plus, it contains several other features making it more robust. The security on Windows NT allows a network administrator to not only provide passwords for resources but also to individuals or groups. This operating system does require the use of a more powerful server computer whose sole function is to act as administrator of the NOS program. Having a server where all access data is stored makes managing Windows NT Server Operating Systems efficient.

Windows NT Server has more than one level of security beyond access/no access. This NOS offers:

· No access.

·Access that restricts the user to read only capabilities.

· Access that allows read and writes usage.

· Access that allows you to change access permissions for network users.

4. UNIX/LINUX

UNIX is the oldest network operating system [3] still being widely used today. It can be used on either peer-to-peer or client/server networks. LINUX is basically a free version of UNIX, which is developed cooperatively by a community of expert programmers. Various computer manufacturers have released proprietary versions of UNIX which run on their particular hardware platforms. Because of differences in processor hardware and variations from UNIX standards, each company's operating system has its own "flavor". End user applications for UNIX must be individually compiled for each different hardware platform and operating system variant. In spite of the rise in popularity of Web and Internet services deployed on Windows NT, for security and reliability UNIX is a popular choice in commercial and university environments. Networking under UNIX is based on the TCP/IP protocol, which has only been integrated into the Microsoft Windows network operating systems in recent years. The early work, which established the Internet, was based on UNIX platforms. Most of the TCP/IP services used in UNIX are also commonly used on1.

1. **Network File System (NFS)**—NFS provides hard disk sharing over TCP/IP networks. It is the primary file and directory sharing protocol used in UNIX. NSF clients are available for Microsoft Windows operating systems, but are generally used for interoperation with UNIX hosts.

2. **Remote login services (RLOGIN and TELNET)**—UNIX has always been a truly multi-user multi-session operating system. Multitasking of applications and processes is extremely robust; much more so than under Windows NT. In addition many users can run private sessions simultaneously under UNIX, by logging in to a host using a character based terminal or emulated terminal over a serial or network connection. RLOGIN and TELNET protocols support logging in over network connections.

3. **Graphical user interface windowing system (X Windows)**—X

Windows is a completely distributed graphical user interface system. Using X-Windows, a user can execute an application on one computer, and let that application interact with a user on a different computer, using a network connection. X Windows allows computers to share their video displays, keyboards and pointing devices with applications running on other computers. There is no requirement that the application execution and display hosts even have the same type of hardware or run the same operating system, as long as the client application and the display server support the X Windows protocol.

5. MAC OS AppleShare

AppleShare provides network services for the Mac OS operating systems. AppleShare supports file and printer sharing over several types of physical networks by using one of the AppleTalk transport protocols: LocalTalk EtherTalk, TokenTalk or FDDITalk. As the Internet has grown in popularity, TCP/IP software has been developed for the Apple MacIntosh computers, along with the standard TCP/IP client applications like Web Browsers and FTP file transfer clients.

AppleShare's peer-to-peer networking is used in small or moderately sized workgroup settings, but has not been adopted widely in large-scale corporate LAN environments.

6. Future Work

In future we will examine all the network operating system with numbers of constant parameter and compare with each others, in case of performance, reliability, throughput and efficiency.

7. Conclusion

In this paper, we see that there are various types of networking operating system exist in the area of server technology. Among them Unix operating system is used widely. But in few recent years, Novell NetWare operating system is gaining its popularity.

8. References

1. http://www.novell.com/documentation/lg/ncs6p/native6.pdf.
2. Tanenbaum, Andrew S., *Modern Operating Systems*.
3. Ritchie, D. M., and K. Thompson, "The UNIX Time-sharing System", in *Bell System Technical Journal,* 57(6):1905-1929. July-August, 1978.
4. Gien, Michel, "Micro-Kernel Design", in *Unix Review,* 8(11):58-63. November, 1990.
5. Clark, David D., "The Structuring of Systems Using Upcalls", in *Proceedings of the 10th Symposium on Operating Systems Principles.* December, 1985.