FILE SECURITY SYSTEM

PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the Degree of Master of Science in applied Science - Software Engineering of the Bharathiar University, Coimbatore.

Submitted by
Mr.T.Senthilkumaran
Reg No.9937S0093

Under the Guidance of
Mrs.Devagi, B.E., M.S.,
Lecturer
Kumaraguru College of Technology
Chinnavedampatti, Coimbatore.

Department of Computer Science of Engineering
Kumaraguru Collegeof Technology.

Coimbatore-641 006.

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore - 641 006.

CERTIFICATE

This is to certify that the project work entitled

"FILE SECURITY SYSTEM"

P-8/6

has been submitted by

MR. T.SENTHIL KUMARAN 99se24

in partial fulfillment of the award of the degree of
Master of Science in Applied Science - Software Engineering
of Bhaarathiyar University, Coimbatore
during the the

Acedemic year 2002-2003

26/9

Head of the Department

26/9

Guide

Certified that the candidate was examined by us in the Project Work Viva Voce Examination held on 26.09.2003 *and the Univerity Register Number was* [illegible]

26/9

Internal Examiner

External Examiner

NCCT
port Software Division

NCCT Ltd. 2nd Floor, Bombay [illegible]
109, Nungambakkam High Road
(Above IOB, Near Ganpat [illegible])
Chennai-34. ✆ : 828 [illegible]
E-mail: ncct@eth.net
URL: www.ncctchennai[illegible]
www.[illegible]

Date: September 24, 2002

To Whom It May Concern

This is to certify that **Mr. T. Senthil Kumaran** has completed his project work successfully in **'C' language** entitled **"File Security System"** on the partial fulfillment of his **M.Sc (Software Engineering)** degree, for a period of three months starting from July 2002.

During this period he had shown an exemplary attitude and approach. Wishing him the very best for ever.

With Regards,

For NCCT

Authorized Signatory

N. C. C. T. ★
CHENNAI.

Acknowledgment

At the outset I would like to thank my Principal **Dr.K.K.Padmanaban, B.Sc., M.Tech. Ph.D., (Engg.)** Kumaraguru College of Technology, Coimbatore, Tamil Nadu, for is constant encouragement throughout my course.

I wish to thank Prof **Dr.S.Thangasamy,B.E., (Hons) Ph.D.,** Head of the Department – Computer Science and Engineering for giving me an opportunity to do this project as a part of my course.

I express my sincere gratitude to **Prof. Mrs.S.Devaki B.E., M.S.** Course Co-ordinator [Software Engineering] and who have personally been my mentor and guide for the successful completion of the project.

I thank my beloved Parents, Sisters and my friends who have extended their encouragement and help physically and mentally to complete and bring out this project successfully.

CONTENTS

INTRODUCTION

The concept of networking has really caught on, especially in the last couple of years. We can many multinational as well as corporate companies going in for networking. The answer to this mass appeal for networks lies in the mere fact that couple of PCs Interconnected provide more or comparable amount of computational as well as overall Efficiency when compared to a powerful and costly mainframe.

Some of advantage of going in for a networking environment for are

1. resource sharing
2. reliability
3. cheaper
4. scalability
5. efficiency

All the above mentioned features of the networking are technical and are highly appealing to the trade people. But there is yet another feature (perhaps the most important) which makes networks a favorite for all the classes in the society.

It is the ability to communicate with virtually any one the world in a very cost effective, reliable and of course in a quicker way. This one aspect of the networks comes in handy to both technical as well as non technical people.

As networks grew, many software vendors have come up with users. And some of the utilities display the message to users by popping them on the foreground screen and there by interrupting the foreground process. Besides not all the incoming messages many be of importance to the user and the chances of the possibility Increases in the presence of michiefmakers who does it deliberately.

Unfortunately not all the network operating system developers took sufficient measures to safeguard the Users from the misuse of theses messaging utilities it is here that this software comes in handy. This software takes care of all the messages hat a user might receive while he is working. This allows the user to concentrate on his work without being disturbed.

PROBLEM DEFINITION & METHODOLOGY

As the networks expanded and flourished so did the supporting software and utilities. There are large number of software's available in the market designed for the users in a Novell NetWare based LAN to communicate with each other by Sending messages across the networks.

Such messaging utilities have one inherent problem, which at times can be quite irritating and disturbing to the user. The problem associated with these utility lies in he ill timing of message display. A user who is busy with his work will be disturbed by the sudden and abrupt popping up of messages. This process Disturbs his foreground and thereby his work.

The proposed software system is aimed at eliminating this problem and there by providing the user with an uninterrupted foreground. The software realizes this by keeping track of the particular interrupt responsible for the messaging process. A different ISR is coded for this interrupt so as to transfer all the messages to the background. These messages are later retrieved and displayed to the user when he is no more busy.

The software can be of help to various concerns and institutions operating in LAN based environments. The user can be relieved of the disturbance and over head associated with normal message reception as well as deliberate attempts from various users to cause trouble.

The entire project is divided in to three modules and are further divided into various submodules.

The main module involved are

Module1 : **NETWORK RECOGNISATION AND TESTING**
Module2 : **MESSAGE CAPTURING AND BUFFERING**
Module3 : **MESSAGE RETRIEVAL AND PROMPTING**

NETWORK RECOGNISATION AND TESTING

The modules does various checking and initialization tests which are required for the proper functioning of the program. The first check involved in this module is to find out whether the user's the user is notified and the software terminates. It further checks for the availability of a local harddisk, availability of Novell NetWare and the NCP IPX.

If the latter of the two tests fails then the user is given an appropriate message and the software terminates. This is done because this software is designed for Novell NetWare based LANs and it requires IPX installed.

MESSAGE CAPTURING AND BUFFERING

The module is implemented as a TSR program, which will trap the particular interrupt responsible for the message transfer. When such an interrupt occurs, it first checks whether the user is busy or not. If the user is not busy then the messages are directly transferred to him. Incase he is busy then the message is captured and is buffered. This buffering takes place either in the local hard disk if one is available. If it is buffered then the message is stored in the server.

MESSAGE RETRIVAL AND PROMPTING

This module is also coded as a TSR, which will reside in the user machine and monitor is foreground process. As soon as he user's foreground becomes free this module check whether any messages are queued up for the user or not. If any messages are present in the buffer then the user is prompted and they are displayed to the user.

DEVELOPMENT PROCESS & DOCUMENTATION

Requirements And Anaz\Lysis Specification

Analysis On User Requirements

The software “serial interrupts” is developed to satisfy the user requirements with their problems being minimized to a certain extent. The analysis is made on several factors, which are discussed below.

In order to understand the user requirements, many existing systems were studied and export opinion were sought. From all these details the following conclusions were drawn

The major requirements of the users are Safe deliver of the messages that are due to them An uninterrupted foreground Steps were taken to meet the above-mentioned needs.

HARDWARE REQUIREMENT

Server

PROCESSOR	:	**PENTIUM II @ 333MHZ**
RAM	:	**64MB**
HDD	:	**4GB**

Workstation

PROCESSOR	:	80386 &ABOVE
RAM	:	8MB &ABOVE
HDD	:	2.1GB
VIRTUAL MEMORY	:	32-BIT
FILE SYSTEM	:	32-BIT
ETHERNET CARD	:	32-BIT

SOFTWARE REQUIREMENT

NETWORK O/S	:	NOVELL NETWARE VER3.12
OPERATING SYSTEM	:	MS-DOS VER6.22
NCP	:	IPX
LANGUAGE USED	:	TURBOC

NETWORKING ENVIROMENT

INTRODUCTION TO NETWORKING ENVIRONMENT

During the twentieth century, the key technology has been information gathering processing and distribution. Among other developments, we have the world wide telephone networks. During the first two decades computer systems were highly centralized, usually with in a large room. The merging of computers and communication has had profound influence on the way computer systems are organized. The old model of a single computer serving all of the organization's computational needs, is rapidly being replaced by computer networks.

WHAT IS THE COMPUTER NETWORK?

A computer network is one, which a large numbers of separate but interconnected computers do the computational work. Two computers are sqid to be interconnected if they are able to exchange information.

NETWORK STRUCTURE

Transmission lines move bits between machines. The switching elements are specialized used to connect two or more transmission lines. When data arrive on a incoming line the switching element must choose an out going line to forward them on. Again following the original must choose an outgoing line to forward them on. We will call the switching elements as **IMPS (INTERFACE MESSAGE PROCESSOR)**

Broadly speaking there are two types of designs for the communication subnet.

1. Point-to-Point Channels
2. Broadcast channels

In the first one, the network contains numerous cables or leased telephone lines, each one connecting a pair of **IMPS**. If two **IMPS** that so do not a cable but communicate indirectly via other **IMPS**, the packet is received at each intermediate **IMP** to another via one or more intermediate **IMPS**, the packet is received at each intermediate **IMPS** in its entirely, stored there until the required output line is free and then forwarded. A subnet using this principle is called a Point-to-Point, store and forward, or packet-switched subnet.

Broadcast systems generally also allow the possibility of addressing a packet to all destinations by using special code in the address field. When a packet with this code is transmitted, it is received and by processed by every machine, something known as multicasting. A second possibility is a satellite or ground radion system. Each IMP has an antenna throughwhich it can send and receive. All IMPS can hear the output from satellite and in some cases.

They can also hear the upwards transmission of their fellow IMPS to the satellite as well. A third broadcast system is the ring. In a ring each bit propagates around on its own, not waiting for the circumnavigates the entire ring in the time it taken to transmit a few bits, often before the complete packet has even been transmitted. Broadcast subnets can be further divided in to static and dynamic, depending on how the channels is allocated. A typical static allocation would be to divide time into discrete intervals and run a round robbin allowing each machine to broadcaster only when its time slot comes up.

There are three types of network depending upon the area under which they are laid. They are,

1. **LAN**
2. **WAN**
3. **MAN**

LAN

If a networks is laid within the same room or with in the same building then the network is said to be a LAN (Local Area Network). LANs generally have three characteristic features.

1. A diameter of not more than a few kilometers.
2. A total data rate of at least several Mbps.
3. Complete ownership by a single organization.

WAN (WIDE AREA NETWORK)

WAN is contrast typically span entire countries have data rates below 1Mbps and are owned by multiple organization.

MAN

MAN is in between LAN and the WAN. MAN is Metropolitan Area Network. A MAN is network that covers an entire city, but uses LAN technology.

LAN TOPOLOGIES:

Topology can be defined as geometric arrangement of workstation and the links among them

Some of the commonly used topologies are:

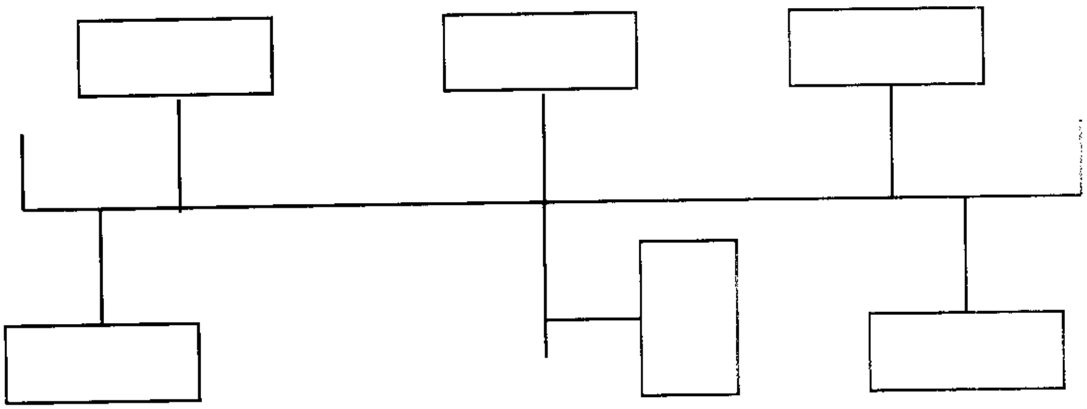
1. **Bus Topology**
2. **Ring or Loop Topology**
3. **Star or Radial Topology**

BUS TOPOLOGY

This consists of single length of transmission medium. This topology is used in traditional data communication. Networks where the hosts at one end of the bus communicate with several terminals attached along its length.

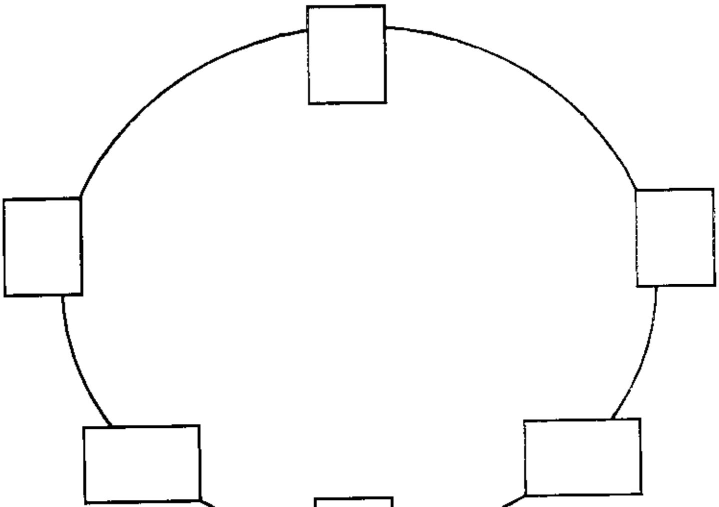
This configuration is known as multi-drop line, which is also the topology used in Ethernet LAN.in bus Configuration all work station are connected to a single shared communication link through interface units and Cable taps. In order to receive a transmission the workstation must be able to recognize their own

Devices attached to a bus there for must posses a high degree of intelligence provided by the Bus Interface Unit.



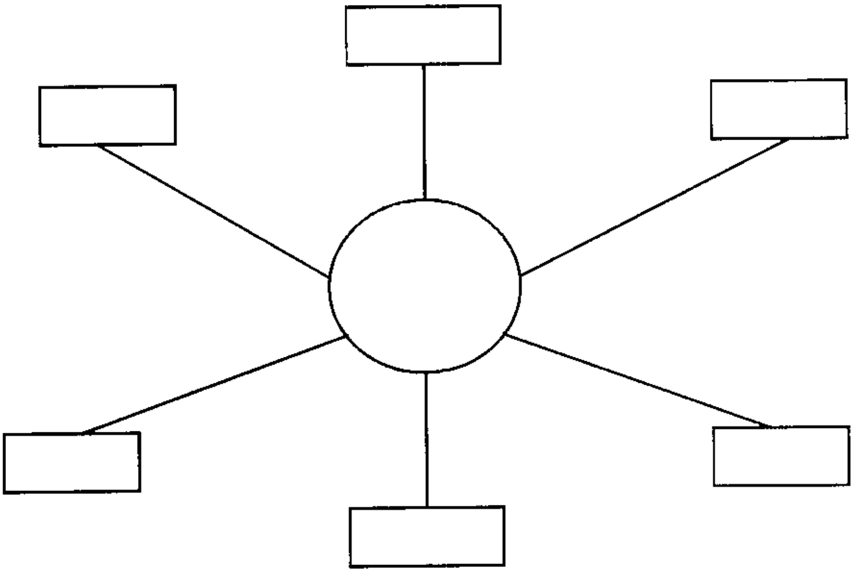
RING TOPOLOGY

In the case of ring or loop topology each node is connected to two and only neighboring nodes. Data is accepted from one neighboring node and is transmitted onwards to another. The data travels in one direction only from node to node around the ring. After passing through each node it returns to the sending node which removes it.



STAR OR RADIAL TOPOLOGY

In the star configuration, each workstation is connected to central server through a dedicated point – to – point channel. Messages are passed from a workstation to the server. Control resides central server or it may be outlying workstations or may be distributed equally to all workstations. If the server fails the whole Network stops.



○ ⟶ **SERVER**

▭ ⟶ **WORK STATION**

TYPES OF LANs

LANs come into two basic flavors: collision- sensing and token passing. Ethernet is is an example of a collision –sensing network and token ring as an example of at token –passing network in the collision. Sensing environment of ten environments often referred to with the abbreviation CSMA/CD (Carrier Sense. Multiple Access, with Collision detection), a network adapter Card with a frame to send first listens to the network to see if it is quiet. If the adapter card hears another card sending a frame at that moment, it waits a microsecond or so and tries again. Even with approach Collisions can and do happen. CSMA/CD networks are designed to expect Collisions and to handle them by retransmitting frames when necessary.

These retransmissions are handled automatically by the adapter card and are transparent to the user. Although poor CSMA/CD network performance is often mistakenly blamed on the number of users simultaneously sending or receiving message traffic on the network, more than 90 percent of transmission problems on an Ethernet network are actually caused by faulty cables or malfunctioning adapter cards.

A token passing network on the other hand can logically be viewed as a ring. This is true although the network can be wired electrically as a star, because data move around the network from workstation to workstation. Each network adapter card regenerates the signal from its "upstream" neighbor and passes the result along to the next workstation. A token is a special type of frame. It

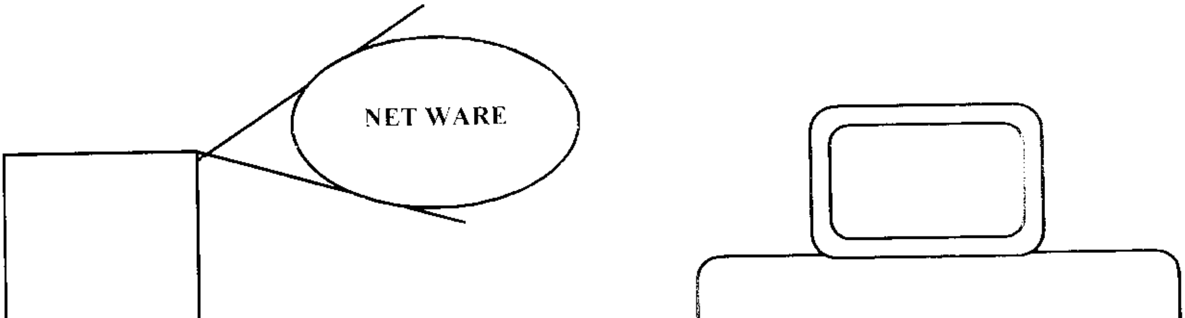
Disk access speed is determined by a number of factors:

- The recording method
- The type and on-board intelligence of the controller
- The type of hard disk(stepper band or voice coil)
- The interleave factor
- The location of the files on the disk (location affects how far the read – write head has to move to get to the file)

FILE SERVER CPU, MEMORY AND NET WORK ADAPTER

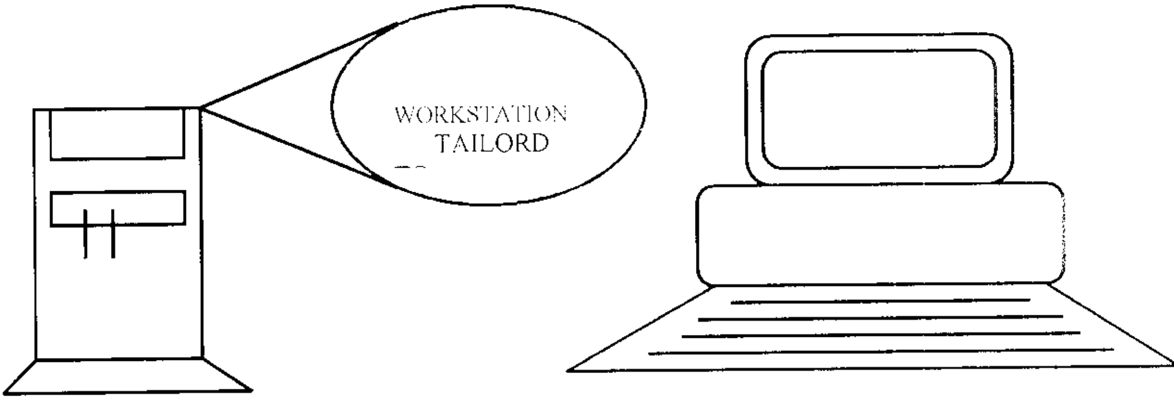
After capacity and speed, the other considerations for the file server are CPU speed, amount of memory, network adapter speed and dedicated versus non-dedicated use. We can achieve significantly improved performance with faster CPU and a sample RAM, which make possible something called catching. Some adapter cards have bigger buffers than others, and thus can hold more frames at one time. Such cards are ideal candidates for installation in a file server. The operating system run on the server is called network operating system (NOS) and has the entire control over the server activities.

SERVER RUNS ON NETWORK OPERATING SYSTEM

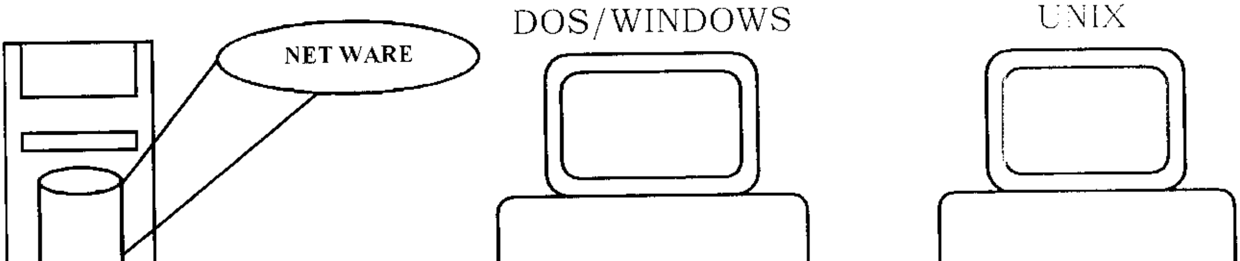


WORKSTATIONS

The workstation are the nodal computers on which the user to there work the workstations on a LAN are personnel computers. These could be IBM PC Compatibles PCs, XTs, ATs etc., workstation on a LAN must have their own processor. At all times, they should have a Network Interface Card (NIC) and proper identifications. Workstations are some times referred to as clients and can run on a variety of platforms with different utilities tailored for each user.



The workstation can run different types of applications in different kinds of environments, provided these platforms are supported by NOS.Net Ware supports Unix, Macintosh, Windows NT and Dos workstations. The most pou



NETWORK SUPPORT SOFTWARE

The network support software must

- Provided access to the network adapter card
- Enable PC-to-PC communications
- Emulate DOS so that the file server's disk are available to the applications running an the workstations.

PC TO PC COMMUNICATIONS

Many vendors have created their own proprietary protocols for peer-level communications on LANs. Regardless of how each particular vendor's protocol is designed, certain basic functions and features are common:

INITIALIZING COMMUNICATIONS:

Each protocol provides the means to identify workstation by name, by number or both. This identification scheme is made available to both the shell/re director layer and to an application. Point-to-point communications are activated by one workstation identifying a destination workstation (often a file server) with which it wants to carry on a dialog.

SENDING AND RECEVING DATA:

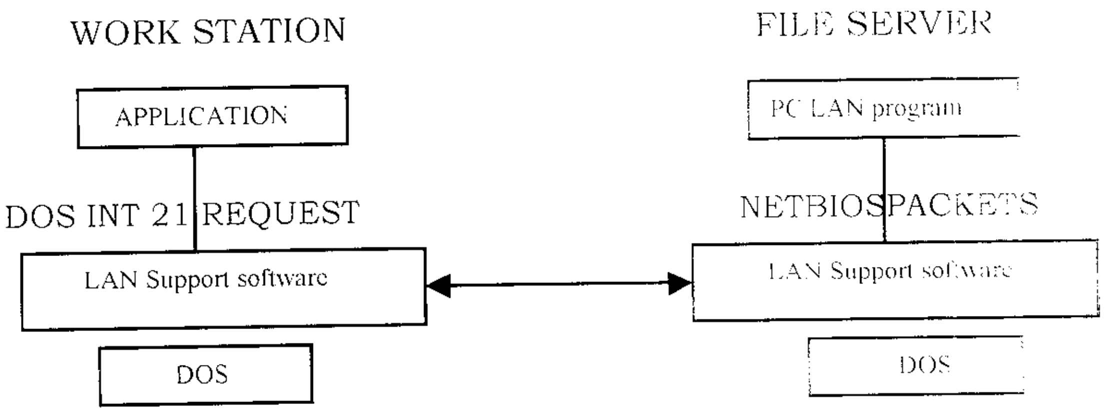
Each protocol provides the means for originating and destination workstations to send and receive message data. A

TERMINATING COMMUNICATION:

The protocol provides the means for the participants to gracefully end a dialog.

DOS REDIRECTION AND EMULATION:

Redirection of DOS function calls makes file-server access possible an application running on a workstations goes through the motions of asking DOS for some part of disk file, but the network software intercepts an application running on a workstation goes through the request and sends it to the file server. The file server does the actual disk I/O to obtain that part of the disk file and returns the result to the workstation. The network software on the workstation hands the disk file contents to the application, and in doing so makes it look as though the local copy of DOS had been one the obtain the file Contents



REDIRCTION OF DOS FUNCTIONS

The network software performs several steps to send the request to the server and get back the response. The first thing the network software needs to do is determine whether it should handle a DOS file read request or pass it along to DOS. It does this by noting at file create or file open time whether time a network drive letter is an effect for the open or create. Call. Because the network software maintains an internal table of which drives are network devices, it is fairly easy for the software to know whether an open or create applies to a network drive. As the file is opened or created the network software assigns a file a file handle or, if the old style file control blocks are being used, it remembers the address of the FCB. When a file read call occurs, the network software examines the file handle or FCB address to know whether the request should be shunted across the network to the file server or passed along for DOS to handle. After receiving the response from the file server, the workstation reverses the steps it took in sending the file read request. The network adapter processes the frames containing the response, the frames headers and frame trailers are stripped off, and the shell/redirector emulates DOS by putting the file data in to the application's buffer, setting the CPU registers to indicate the number of bytes actually read, and returning to the application at the next instruction after the DOS function call.

NETWORKING OPERATING SYSTEMS

"You can virtually cruise through the world" Unifying all the network components is the Net work Operating System. The Network Operating System is the software facilitates file, printer sharing, and communication between workstations, such as FILE TRANSFER.

This is only software component of the LAN and is extremely important. The features of the Network Operating System has a direct bearing on the performance of the LAN. The popular Network Operating Systems in use are NetWare series form.
NOVELL, IBM's PCLAN, 3+Open from 3com and Vines from Banyan systems.

Each have their own distinctive features. For example. Vines offers a lot of facilities when it comes to WAN and connectivity solutions. But for an overall performance in the LAN area, NOVELL NetWare has becomes very popular and they have over 50% if the market.

The network operating system consists of two distinct portions. The first portion is loaded in to the memory of the workstations and resides as shell above the workstations own Operating Systems. This "Shell" intercepts all requests from the workstation and determines whether it is a Network request or not. The shell directs local commands to the workstations own operating system and redirects network commands to the file server. The strength of a good Network Operating System is the transparency of its services to the users. The application software should work as well as it does on a stand-alone microcomputer while adding the benefits of the Local Area Network.

NOVELL NETWARE

Netware is a very popular network operating system used today. More than half of all the LAN installation runs versions NetWare. It is a favorite in the market because of its efficiency, versatility and security. It is versatility is evident from the fact that it runs well on several different kinds of hardware. Netware also provides a rich set of services and facilities to the programmers.

CLIENT/ SERVER CONCEPT

The client server concept was introduced by Novell. The heart of a LAN is a file server, which is usually the biggest computer on the net work. They are powered by the most powerful processors and have high memory storage capabilities. The most important purpose of the file server is to store files that are shared by users on the network. Some of the works done by the file server is even invisible to the administrator.

The workstations in NetWare terminology are called clients. They are called so because they request and use the resources available on the server and client is achieved by the various protocols defined by Novell.

NET WARE SERVER ARCHITECTURE

A Net ware file server is really a mini computer in disguise, although it resembles an ordinary IBMATor PS/2. The format of the server file system is completely different from that of PC Dos. This foreign format enables net ware to keep track of files and directories more effectively that is possible under normal PC Dos. In addition to the normal Dos file attributes like **r h a s** etc., the Net Ware provides additional features and attributes like shareable, non shareable etc, it also associates and identification tag with every file, which has information about the date of creation, owner, rights etc.,

Net Ware operates on the server in protected mode and literally takes control of the entire computer. Net Ware also piles up extra memory for catching. Net Ware also supports Value Added Processes (VAPs), which enables to separate program modules link with Net Ware to provide extra service.

NET WARE WORK STATION

Both of the software components running on each Net Ware workstation are TSR programs. IPX manages the PC to PC communication and PC to the file server communication by implementing the IPX/SPXcommunications protocol. NET3 is the shell/redirector that shunts the Dos file request and from the file server by issuing commands to IPX. These components together make file server's disk and peripherals look like Dos managed peripherals. IPX takes about 19-KB memory and NET3 about 38KB.

Net Ware also supports different protocols like NETBIOS in addition to the IPX /SPX protocols. The support for NETBIOS is achieved by a loading a NETBIOS emulator, which essentially acts as a middle man translating all the NETBIOS command to the IPX form for execution. This features essentially a tradeoff between speed, memory and portability. The additional flexibility that a programmer gets is really worth it. Net Ware 286 file server supports as many as 100 workstations. In case of Net Ware 386, the support extended is up to 250 workstations or users. Net Ware 386 also supports as much as 4GB of memory for catching. As much as, 32TB of memory can reside in a single server, as much as 100,000 files can be opened at the same time, and a file can span multiple drives.

WORKSTATION SERVER INTERACTION

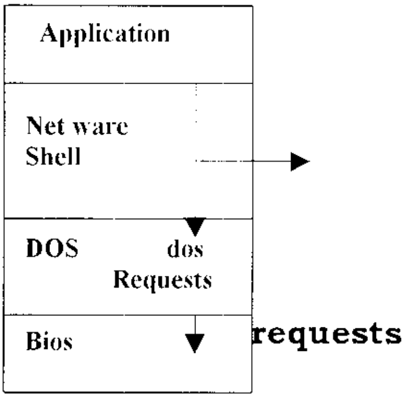
Between the workstation and server, there must be a medium through which the machines communicate. This medium can be as simple as a telephone wire or complex as a satellite link. As mentioned above the server is run by the Network Operating System (NOS) and the workstation is run by the workstation operating system which can be of different types. Net Ware supports DOS, Windows NT< Unix and Macintosh clients. The most popular workstation and a Net Ware server LAN are DOS. In order to achieve the communication between a DOS workstation and a NetWare server, two distinct software components are required they are

1. DOS shell / redirector
2. Open Data Link Interface (ODI)

DOS SHELL/REQUESTOR

This component is responsible for ensuring that the requests for DOS services are directed to the DOS while requests for Net Ware services are sent to the server. Prior to net ware4 and the net ware3.12, the technology used was called the Net Ware DOS Shell. The current technology is called the Dos requestor. Although both of them do the same, service the way they do it different.

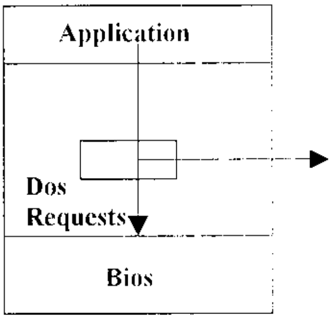
DOS SHEEL



The shell works by intercepting the program structures known as interrupts: the DOS services are serviced as ISRs. The shell works by redirecting the service requests to it self thus ensuring that these services will be processed by the server before passing it to DOS.

DOS REQUESTOR

There is a very big disadvantage with shell approach. As the DOS versions changed, Novell had to write different shell program for each version. The various versions are known as NET2.com, NET3.com NET4.com etc., owning to these problems a new technology named DOS requestor was introduced. It is required for Net Ware 4.0 and is the recommend method for interconnecting DOS workstations.



Network service requests

The way DOS requestors work is quite from the Dos shell. Here Dos bears the responsibility for redirecting the service requests.

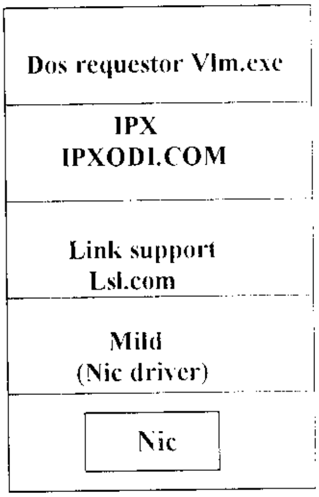
The requestor enables responsibility the applications to access the service in three ways.

1. by direct calling the Dos requestor
2. by means of Dos redirector
3. through a pipeline that bypasses Dos entirely

OPEN DATA LINK INTERFACE

This component is responsible for actually enabling the workstation to interface with the net work cabling. This component must include software driver for the network interface cards as well as support for any protocols that are needed. The standard protocol for net ware is called Internet packet exchange (IPX). Together these drives and protocols are called (ODI).

HOW THEY WORK TOGETHER


Dos requestor Vlm.exe

IPX
IPXODI.COM

Link support
Lsl.com

Mild
(Nic driver)

Nic

Together the Dos requestor and the ODI drivers enable a DOS PC to function as a client on Net Ware LAN. As shown in the figure it is layered architecture. Because the DOS requestor and ODI follows a layered architecture. Because the DOS requestorsand ODI follows a layered architecture, they are very flexible when changes are required. If DOS changes then only the DOS requestor, need to be modified. In the reverse case only IPXODI driver need to be replaced and the overall network remains unaffected.

IPX DATAGRAM SERVICE:

The standard protocol used in Novell Net Ware is IPX which stands for inter network packet exchange. It is a close adaptation of protocol Xerox Network Standards (XNS) it supports only data gram messages. This protocol perform addressing, routing and switching to deliver a message to its destination.
IPX although faster than other protocols, is not a safe one for messaging. This is due to its data gram nature which is connection less service. But Novell claims that 95%of the IPX packets are delivered correctly.

IPX DESTINATION ADDRESS:

The destination address in an IPX packet is an Internet work address, which consists three components:

- Network numbers identify each segment of the multi server network.
- System administrators assign the numbers when the Novell Net ware file servers are initialized.
- Node address uniquely identified each Network Adapter card. Your application opens and closes sockets in much the same way as it opens and closes files. For IPX packets, the destination address can contain a group address send IPX packets by specifying the destination address, but you receive them by

IPX PROGRAM SERVICES

PC-to-PC communication is managed by using IPX services. These services rely on 3 data structures that you construct and pass to IPX.

- The IPX header, which consists of 30 bytes of IPX packets.
- The data record or message that you want to send or receive.
- An Event Control Block (ECB) that is not actually transmitted but contains information about a particular IPX operation that you initiate.

IPX PACKET FORMAT

The following figurers shown an IPX packet format:

Check sum	2
Length	2
Transport control	1
Packet type	1
Destination network	4
Destination node	6
Destination socket	2
Source network	4
Source node	6
Source socket	2
Data portion	0-546

DESTINATION

IPX packet is the data record that is placed either inside the Ethernet frame or token ring fame for transmission or reception the fields within an IPX packet are the following:
Checksum: this 2-byte field is a hold over. Because the lower level protocol always performs error checking, you never need to set this field. Length- the size of the complex packet, including both IPX packet header and the data portion, is expressed by this 2-byte field. IPX calculate the value of this field based on information you provide when you tell IPX to send a packet. You do not set this field directly.

Destination Network-this is a 4byte field that identifies the network on which the intended receiver workstation is located.
Destination node – These 6 bytes identify the target workstation by using its unique physical address;

Destination Socket – This field contains a 2-byte socket number. It is the last of the three fields you specify to say where the packet should be delivered. The socket must have been opened by the application running on the destination.

Source Network – The network number of the originating workstation, this 4byte field is set of IPX.

Source node – the physical address of the network adapter card in the originating workstation, this 6-byte field is set by IPX.

Source Socket – the packet is send through on open socket, which IPX sets in this field.

INITIALIZATION AND TERMINATION FUNCTIONS:

When two PC is on the network, want to send message records back and forth using IPX, the application on both workstations first opens a socket using the IPX OPENSOCKET function. The socket number that is open on workstation a must be known to the application B, and vice versa.

Both workstations need to know the destination address of other workstation. Socket numbers are easily determined; you simply establish conventions for the sockets you use.

How ever, your application does not automatically know the network number and node address you know only the user's ID. IPX function sat the beginning of your program. Net Ware lets a single user ID log on at several workstations simultaneously. The usual way to handle this situation is by using the first item from the list and ignoring any subsequent items in this list.

When finished sending and receiving message records, the two workstations close the open sockets by calling IPX CLOSE SOCKET.

SENDING AND RECIVING PACKETS:

After you have opened a socket at workstation A and determined the network number node address and socket number of workstation B, you are ready to send and receive message records. IPXSEND PACKET and IPX LISTEN FOR PACKET. Are used to do the sending and receiving. Each time you call IPXSEND PACKET, you should supply the immediate address field you can call the function IPXGET LOCAL TARGET determine the value to be placed in the immediate address field.

SOCKETS:

One of the first things you want to do in a program that uses IPX is to open a socket. To send a data packet through a socket to its destination; you receive a data packet through a socket. You must avoid using the same socket numbers that NetWare is using. You also need to assign them in such way that your application can easily determine the destination socket number when its want to send a data packet another consideration is that NetWare defines the two bytes of socket number "backward". NetWare expects to see the most significant byte first.

A socket can belong – lived or short –lived. You can close either kind of socket with call to IPX –close socket. A short-lived socket will be closed automatically when a program terminates. Long-lived sockets are used if you write terminated resident programs.

To open a socket put the socket number in the DX register and set AL according to the longevity you want to the socket to have. Set AL to 0xFF for a long lived socket. The function code for open –socket is zero, which goes into BX register. To close a socket, put its number in the DX register and call IPX with a function code of one in BX. If a send or receive IPX operation is outstanding when you close the socket, IPX cancels that operation.

SESSION LEVEL COMMUNICATION USING SPX

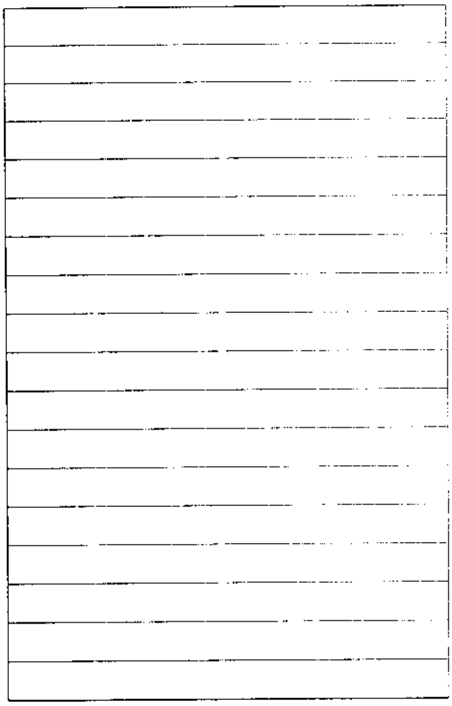
Sequenced packet exchange is a session level connection oriented protocol before SPX packets are send or received, a connection must be established between the two sides can exchange information with guarantee that the message will be received. SPX also guarantees the sequenced delivery of messages. SPX operates at a layer Above the IPX. SPX actually makes use of the IPX protocol for message delivery. The function SPX initialize which is used to determine whether SPX is available or not.

In order to communicate with each two machines should first initialize SPX using SPX initialize call. Each workstation then open one or more number of sockets by issuing IPX open socket command. Then the connection is established by the function calls SPX establish connection and SPX listen for connection.

If the SPX establish connection calls is issued first and if all the retry attempts performed by SPX are exhausted before the other station issues an SPX listen for connection then the

SPX PACKET FORMAT

SPX PACKET FORMAT



IBM NETBIOS

NETBIOS is IBM's protocol for PC-to-PC transfer of messages and records however, Novell does provide a NETBIOS emulator with NETWARE. Because it uses a different frame format. Novell's is NETBIOS software is in compatible with the IBM NETBIOS program product. NETBIS which stands for network basic input/output system is the protocol that the IBM PC LAN program uses to send file – service requests back and forth to the file server. The protocol's communication facilities are of course also available to the program you write NETBIOS corresponds to the network layer, transport layer, and session layer of the OSI model; the protocol operates at a higher level than either IPX or SPX.

communications IBM PC LAN program file service packets send to and received from a file server are managed under session control rather than treated as individual diagrams. This is one of many differences between the IBM PC LAN program and Novell NetWare. Most of the NETBIOS commands come in both wait or no-wait flavors. When we use the wait version of a command, NETBIOS completes the operation before returning to your program. If you specify no –wait you then have the option of polling or giving NETBIOS the address of one of your routines that NETBIOS will invoke when the command is completed. This facility is simulating to the IPX/SPX event service routine concept.

NETWORK CONTROL BLOCK (NCB'S):

To invoke a particular NETBIOS command, your application builds a Network Control Block (NCB) and then executes an Interrupt 5C(hex.). the figure shows the format of an NCB, and this section describes each of the fields:

You set the 1-byte command field to tell NETBIOS, which command you, want it to execute. If the higher order bit is set, the command is execute in no-wait mode.

The 1-byte return code field contains the immediate error code (set by NETBIOS when it begins executing the command). After a Listen or Call command is executed, the 1-byte local session number field contains the LSN assigned to that session. For Send or Receive commands, you put the session's LSN in this

NETBIOS returns the 1-byte name number field after an add name or add group name command. You use this number, not the name, when doing any data gram related commands or the receive any command.

In this 4-byte (segment: offset) data buffer address field, you put a far pointer to the data buffer associated with a send or receive a operation.

You set the 2-byte data buffer length field to indicate the length of the data buffer.

You set the 16-byte call name field to indicate the name of the other workstation with which you want to communicate.

You set the 16-byte local name field to indicate by which of the names in the local name table (or the permanent node name) you want your application to be known.

When you issue a call or listen command, you set the 1-byte receive time out field to value that indicates how many half-second intervals NETBIOS should use while waiting for a subsequent receive command to be completed. A value of 0 indicated no time-out.

You put into the 4-byte POST routine address a far pointer (segment: offset) to a routine that NETBIOS invokes when the

WHY IPX?

We have chosen to make use of IPX services in this project. The advantage of going for IPX is very high when compared to other protocols. IPX being the standard protocol of NetWare based systems, helps the programmer in developing efficient and reliable Utilities than those of other protocols.

Net Ware provides support for third party protocols like IBM NETBIOS. Although this facility adds to the portability and flexibility of the system it has it is own, disadvantages are due to the fact that the NETBIOS emulator that is to be loaded poses the following problems.

1. The emulator software takes up memory up to 25K
2. The NETBIOS commands should be translated to the underlying IPX commands for execution. This translation reduces the speed and increase the processor overhead.
3. These problems are eliminated by using IPX services.

TSR

TSRs stands for Terminate and Stay Resident programs the sole reason why TSR exist today is because designers of MS-DOS failed to foresee the need for single task-switching between concurrently resident programs.

An ordinary DOS program, on execution is loaded from the

MODULE3:

MESSAGE RETRIEVAL AND PROMPTING

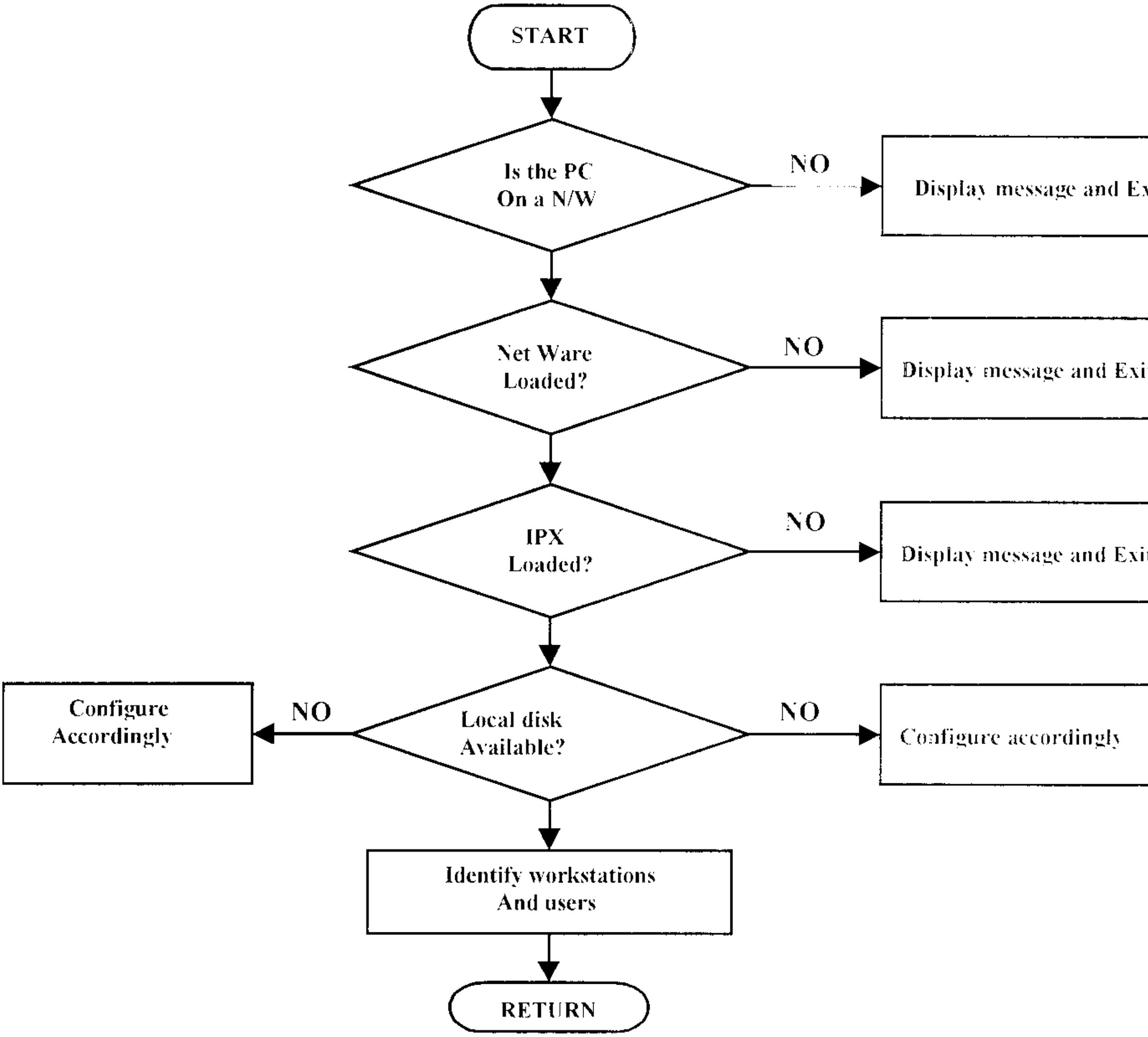
This is responsible for retrieving the messages of a user if there are any and prompting him. The main objective here is to provide the user with an uninterrupted fore ground. This is done by diverting the messages sent by various popup utilities so that the user is not interrupted. This was achieved by the second module. The next step is to retrieve the messages that might be in store for the user. In order to realize this function, this module is also implemented as a TSR program. This program takes care of the above mentioned objectives as follows.

1. It resides in the user machine and constantly monitorsa predefined location for any messages.
2. If need some messages are in store then
   2.1 The foreground activity is monitored and in case it is free the user is prompted about the messages and the message is displayed
   2.2 In case of fore ground activity, the program enters a wait state ready to trigger as soon as the fore ground is free.
3. If there are no messages then the program enters a waiting state waiting for the arrival of messages.

The monitoring for the messages are carried out by capturing the timer interrupt. Thus, the program checks for the arrival of messages in a well known predefined location once in every 20 seconds. The PC timer ticks 18.1 times in a second. Th

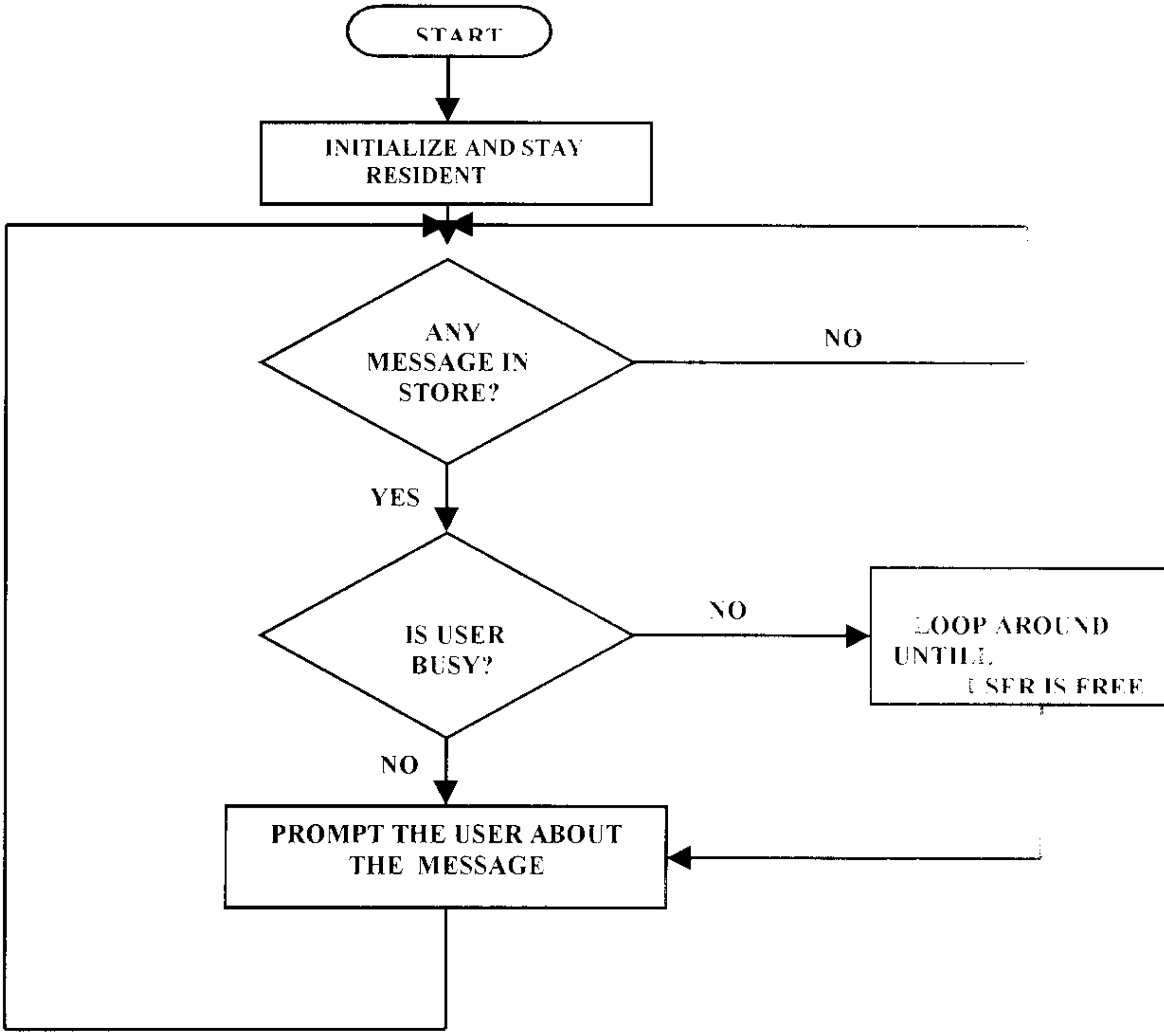
FLOWCHART

MODULE1: NETWORK RECOGNISATION AND TESTING



MODULE 3:

MESSAGE RETRIEVAL AND PROMPTING:


START
INITIALIZE AND STAY RESIDENT
ANY MESSAGE IN STORE?
NO
YES
IS USER BUSY?
NO
LOOP AROUND UNTILL USER IS FREE
NO
PROMPT THE USER ABOUT THE MESSAGE

IMPLEMENTATION:

The project implemented using TURBO C for Net Ware based LAN environments recognition and testing is implemented issuing various interrupt calls that are defined by Net Ware as well as DOS.

The second and third modules namely,

1. Message capturing and buffering.
2. Message retrieval and prompting

Are implemented as TSR programs. They are TSR because they are required to monitor the occurrences of various events.

The TSR concept is realized by making use of standard function called KEEP () which is used to make a program resident. This function requires two parameters which are used for determining the exit status and the amount of memory required

The constant monitoring of events are achieved by hooking into the interrupts that are responsible for the occurrence of such events. This process in effect replaces the original ISRs of the interrupts with new event handlers. These new event handlers do the required work to achieve the message capturing buffering and retrieval.

The hooking interrupts are achieved by inserting the address of new event handlers in the place of old ones. This process is done

(Interrupt Vector Table) is the portion of the memory where the address of the ISRs of various software uninterrupted are stored.

These function can be achieved either through far pointers and pointers to functions, or using the TURBO C function GETVECT () and SETVECT () both these functions require parameters which gives them the full details about.

CONCLUSION

REFERENCE

BOOK: The Complete Reference C
Third Edition ,Herbert Schildt.

BOOK:Programming with c
Second Edition,Byron Gottfried.

WEB ADDRESS:

www.code gruru.com.

www.enctyption.com

www.decryption.com