

TABLE OF CONTENTS

TABLE OF CONTENTS	1
PREFACE.....	3
ACKNOWLEDGEMENT.....	4
INTRODUCTION TO LINUX.....	5
WHAT IS LINUX?	5
FEATURES OF LINUX	5
INSTALLING LINUX	7
HARDWARE REQUIREMENTS	7
IMPORTANT DIRECTORIES IN THE LINUX FILE SYSTEM	9
/	9
/home.....	9
/bin	9
/usr	9
/usr/bin.....	9
/var/spool	10
/dev.....	10
/usr/sbin.....	10
/sbin.....	10
/etc.....	10
INTRODUCTION TO SAMBA	11
WHAT IS SAMBA?	11
WHAT CAN YOU DO WITH SAMBA?	11
INSTALLING SAMBA.....	12
CONFIGURING SAMBA.....	12
STARTING AND STOPPING THE SAMBA SERVER	13
smbd.....	13
nmbd.....	14
SMB.CONF FILE	14
CONFIGURATIONS	21
GLOBAL SETTINGS: [GLOBAL]	21
# Workgroup	21
# NetBIOS Name.....	21
# Server String	21
# Hosts Allowed	21
# Loading Printers	22
# Log files.....	22
# Security modes	22
# Password Levels.....	23
# Password encryption.....	23

<i># Sockets Options</i>	24
<i># Remote Announcement</i>	24
<i># Browser Control Options</i>	24
# LOCAL MASTER.....	24
# OS LEVEL.....	24
# DOMAIN MASTER.....	24
# PREFERRED MASTER.....	25
<i># Domain Logon Server</i>	25
<i># Name Resolve Order</i>	25
<i># Windows Internet Name Serving (WINS) Support</i>	25
<i># DNS Proxy</i>	26
SHARE DEFINITIONS:	26
<i>Home Directories [homes]:</i>	26
<i>Profiles: [profiles]</i>	26
<i>Share Permissions: []</i>	27
<i>Printing: [printers]</i>	27
SUMMARY	28

PREFACE

The main aim of doing this project is to provide connectivity between Windows NT and Linux machines. For this purpose, software Samba Server is used. We tried our best to perform the complete functionality in this concern. The idea is that the user can view their NT files on the Linux machine and vice versa.

ACKNOWLEDGEMENT

We thank our GOD, who helped us in every field of life to achieve our objectives. We would like to thank our respected teacher *Mr. Sajjad Ali*, who guide and help us at every step for completion of this project. We would like to thank Mr. Usman Saleem who helps us in learning of different commands of Linux. And finally, we would like to thank our friends who helped us in completion of this project.

Fahad Habib
Baber Amin
Bilal Rasool
Shahbaz Sarwar
Bilal Aziz

INTRODUCTION TO LINUX

What Is Linux?

Linux is a free; UNIX work-alike designed for Intel processors on PC architecture machines. Linux is not UNIX, as UNIX is a copyrighted piece of software that demands license fees when any part of its source code is used. Linux was written from scratch to avoid license fees entirely, although the operation of the Linux operating system is based entirely on UNIX. It shares UNIX's command set and look-and-feel, so if you know either UNIX or Linux, you know the other, too.

Linux supports a wide range of software, from TeX (a text formatting language) to X (a graphical user interface) to the GNU C/C++ compilers to TCP/IP networking. Linux is also compliant with the POSIX.1 standard, so porting applications between Linux and UNIX systems is a snap.

Features of Linux

Some of the important features of Linux that make it so unique are:

- **Full multitasking and 32-bit support.** Linux, like all other versions of UNIX, is a real multitasking system, allowing multiple users to run many programs on the same system at once. Linux is also a full 32-bit operating system, utilizing the special protected-mode features of Intel 80386 and later processors and their work-alikes.
- **The X Window System.** The X Window System is the de facto industry-standard graphics system for UNIX machines. A complete version of the X Window System, known as XFree86, is available for Linux. The X Window System is a very powerful graphics interface, supporting many applications.
- **TCP/IP (Transmission Control Protocol/Internet Protocol) support.** This is the set of protocols that links millions of university and business computers into a worldwide network known as the Internet. With an Ethernet connection, you can have access to the

Internet or to a local area network from your Linux system. Using SLIP (Serial Line Internet Protocol) or PPP (Point to Point Protocol), you can access the Internet over phone lines with a modem.

- **Virtual Memory and Shared Libraries.** Linux can use a portion of your hard drive as virtual memory, expanding your total amount of available RAM. Linux also implements shared libraries, allowing programs that use standard subroutines to find the code for these subroutines in the libraries at runtime. This saves a large amount of space on your system; each application doesn't store its own copy of these common routines.
- The Linux kernel uses no code from AT&T or any other proprietary source. Much of the software available for Linux is free. In fact, a large number of utilities in Linux are developed by the GNU project at the Free Software Foundation in Cambridge, Massachusetts. However, Linux enthusiasts, hackers, programmers, and recently even commercial companies from all over the world have contributed to the growing pool of Linux software.
- Linux supports (almost) all of the features of commercial versions of UNIX. In fact, some of the features found in Linux may not be available on other proprietary UNIX systems.
- **GNU software support.** Linux supports a wide range of free software written by the GNU Project, including utilities such as the GNU C and C++ compiler, gawk, groff, and so on. Many of the essential system utilities used by Linux are GNU software.
- Linux is compatible with the IEEE POSIX.1 standard. Linux has been developed with software portability in mind, thus supporting many important features of other UNIX standards.
- **Built-in support for networking, multitasking, and other features.** You'll see this touted as "New Technology" in systems such as Windows NT. In fact, UNIX (and now, Linux) has implemented this "new technology" for more than 15 years.

- **Linux is cheaper** to get than most commercially available UNIX systems and UNIX clones. If you have the patience and access to the Internet, the only price you pay for Linux is your time. Linux is freely available on the Internet.

The **Most Important Advantage** of using Linux is that you get to work with a honest-to-goodness kernel. All of the kernel source code is available for Linux, and you have the ability to modify it to suit your needs. Looking at the kernel code is an educational experience in itself.

Installing Linux

The process of installing Red Hat Linux is straightforward, although there are lots of little problems scattered throughout the process that can cause hassles. Red Hat Linux is much easier to install than many other versions of Linux, so you will find the installation goes quite easily. The Red Hat installation program handles common errors that you might make, and the system is versatile enough to handle most configuration requirements.

The essential steps for installing Red Hat Linux are as follows:

1. Create boot and root disks.
2. Boot Linux from floppies or CD-ROM.
3. Partition the hard disk.
4. Create a swap file.
5. Create a Linux file system.
6. Install the Linux software.
7. Configure the kernel.
8. Set the boot process.
9. Reboot into Linux from your hard disk.

Hardware Requirements

The following is a rough guideline of some hardware requirements for Linux. You do not have to follow them directly, but this list should give you a rough idea of what's required.

- An Intel 80386 or better CPU (the faster and more powerful the better, of course). You don't need a math coprocessor, although it's strongly recommended as it speeds up a lot of graphics operations, especially under X. If you have an 80386 chip, 80387 math coprocessors are available separately and are installed in a socket on your motherboard. If you have an 80486 processor, the math coprocessor is on the 486 chip itself. Pentium and Pentium Pro CPUs have the coprocessor built in.

If you don't have a math coprocessor, the Linux kernel will emulate floating-point math for you. If you do have one, however, floating-point math will be handled by the hardware, which for some applications is a real plus.

- Your system must have an ISA, EISA, PCI, or local bus architecture machine. These terms specify how the CPU communicates with hardware, and are a characteristic of your motherboard. Most existing systems use the ISA and PCI bus architecture.
- At least 4MB of RAM should be available. Memory is speed, so if you have more RAM you'll thank yourself for it later. If you're a power user, 8MB should be more than enough for most applications. If you want to run X Window, your system will require at least 8MB of RAM.
- A Hard Drive with space available for installing Linux. The amount of space required depends on the amount of software you're installing and how much free space you wish to leave yourself. You can install Linux in very small amounts of disk space, but a realistic minimum is about 150MB. For a full system with X and development tools, much more is required. The complete installation can use up 250MB, with more useful for data files.
- A Hercules, CGA, EGA, VGA, or Super VGA video card and monitor. In general, if your video card and monitor work under MS-DOS or Microsoft Windows, then Linux should be able to use them without any problem. However, if you're going to use the X Window system (either Metro-X or Xfree86), some video configurations are not supported.

Important Directories in the Linux File System

Most of the directories that hold Linux system files are "standard." Other UNIX systems will have identical directories with similar contents. This section summarizes some of the more important directories on your Linux system.

/

This is the root directory. It holds the actual Linux program, as well as subdirectories. Do not clutter this directory with your files!

/home

This directory holds users' home directories. In other UNIX systems, this can be the /usr or /u directory.

/bin

This directory holds many of the basic Linux programs. bin stands for binaries, files that are executable and that hold text only computers could understand.

/usr

This directory holds many other user-oriented directories. Some of the most important are described in the following sections. Other directories found in /usr include

docs:	Various documents, including useful Linux information.
man:	The man pages accessed by typing man <command>
games:	The fun stuff!

/usr/bin

This directory holds user-oriented Linux programs.

/var/spool

This directory has several subdirectories. mail holds mail files, spool holds files to be printed, and uucp holds files copied between Linux machines.

/dev

Linux treats everything as a file. The /dev directory holds devices. These are special files that serve as gateways to physical computer components. For instance, if you copy to /dev/fd0, you're actually sending data to the system's floppy disk. Your terminal is one of the /dev/tty files. Partitions on the hard drive are of the form /dev/hd0. Even the system's memory is a device.

A famous device is /dev/null. This is sometimes called the bit bucket. All information sent to /dev/null vanishes—it's thrown into the trash.

/usr/sbin

This directory holds system administration files. If you do an `ls -l`, you see that you must be the owner, root, to run these commands.

/sbin

This directory holds system files that are usually run automatically by the Linux system.

/etc

This directory and its subdirectories hold many of the Linux configuration files. These files are usually text, and they can be edited to change the system's configuration.

INTRODUCTION TO SAMBA

What is Samba?

If you look at any English dictionary, Samba is defined as a Brazilian dance, but Samba on Linux is something completely different. Samba is an implementation of a Server Message Block (SMB) protocol server that can be run on almost every variant of UNIX in existence.

Samba is an open source project, just like Linux. The entire code is written in C so it is easily portable to all flavors of Linux. Samba is a tool for the peaceful coexistence of Linux and Windows on the same network on the level of File and Print sharing over the NetBIOS protocol. It allows Linux systems to move into a Windows “Network Neighborhood” without causing a mess. With Samba, Linux servers are acting as any other Windows server, offering its resources to the SMB clients.

What can you do with Samba?

- With Samba, a Linux server can act as a file/print server for Windows networks. It can replace expensive Windows NT file/print servers in this role, creating a less expensive solution.
- Samba can act as a NetBIOS name server (NBNS) in a Windows world referred to as WINS - Windows Internet Name Service.
- Samba can participate in NetBIOS browsing and master browser elections.
- Samba can provide a gateway for synchronization of UNIX and Windows NT passwords.
- With Samba client software you can access any shared directory or printer on Windows NT servers or Samba servers and allow UNIX machines to access Windows NT files.

- Using the Samba File System (SMBFS) you can mount any share from a Windows NT server or a Samba server in your directory structure (this is only available on Linux).

Installing Samba

Whenever you install the Red Hat Linux latest version, it automatically installs the Samba packages. If you want to install it manually than following commands are used:

From Red Hat Linux CD run these commands.

```
mount /mnt/cdrom  
rpm -ivh /mnt/cdrom/RedHat/RPMS/samba*
```

This will install the Samba server, client, and common packages.

Configuring Samba

In this section we will explain how to configure Samba so it can participate as a file and print server in an existing Windows network or be a stand-alone file and print server for Windows and Linux clients.

Before start using Samba, you will need to configure the smb.conf file. This file is the heart of the Samba server. When the Samba package is installed, a default configuration file is installed in:

```
/etc/smb.conf
```

The smb.conf file is divided into two main sections:

1. Global Settings

This section is used to define connection parameters that help in connection to Windows machine.

2. Share Definitions:

Shares are defined in this section. A share is a directory on the server that is accessible over the network and shared among users. This section can be divided into three subsections:

- i. **Homes:**
In this subsection, user's home directories (Profiles) are defined.
- ii. **Printers:**
Available Printers are defined in this subsection.
- iii. **Shares:**
In this subsection, an entry for each share can be added which you would like to define.

Starting and Stopping the Samba server

You can start the Samba server by executing the following command:

```
/etc/rc.d/init.d/smb start
```

You will see:

Starting SMB services:	[OK]
Starting NMB services:	[OK]

Two daemons are started:

- `smbd`
- `nmbd`

smbd

The *smbd* daemon is responsible for managing the shared resources between the Samba server machine and its clients. It provides file, print, and browser services to SMB clients across one or more networks. *smbd* handles all notifications between the Samba server and the network clients. In addition, it is responsible for user authentication, resource locking, and data sharing through the SMB protocol.

nmbd

The *nmbd* daemon is a simple name server that mimics the WINS and NetBIOS name server functionality, as you might expect to encounter with the LAN Manager package. This daemon listens for name server requests and provides the appropriate information when called upon. It also provides browse lists for the Network Neighborhood and participates in browsing elections

Executing the following command can stop the Samba server:

```
/etc/rc.d/init.d/smb stop
```

You will see:

```
Stopping SMB services: [OK]  
Stopping NMB services: [OK]
```

Whenever you modify the `smb.conf` configuration file you must restart the Samba server.

SMB.CONF FILE

```
# This is the main Samba configuration file. You should read the  
# smb.conf(5) manual page in order to understand the options listed  
# here. Samba has a huge number of configurable options (perhaps too  
# many!) most of which are not shown in this example  
#  
# Any line which starts with a ; (semi-colon) or a # (hash)  
# is a comment and is ignored. In this example we will use a #  
# for commentary and a ; for parts of the config file that you  
# may wish to enable  
#  
# NOTE: Whenever you modify this file you should run the command  
# "testparm" to check that you have not many any basic syntactic errors.
```

```
# ===== Global Settings =====
```

```
[global]
```

```
# workgroup = NT-Domain-Name or Workgroup-Name
```

```
workgroup = STUDENT
```

```
netbios name = SAMBASYS
```

```
; netbios aliases = sambasys
```

```
# server string is the equivalent of the NT Description field
```

```
server string = Samba Server
```

```
# This option is important for security. It allows you to restrict  
# connections to machines which are on your local network. The  
# following example restricts access to two C class networks and  
# the "loopback" interface. For more examples of the syntax see  
# the smb.conf man page
```

```
hosts allow = 192.168.1.
```

```
# if you want to automatically load your printer list rather  
# than setting them up individually then you'll need this
```

```
printcap name = /etc/printcap
```

```
load printers = yes
```

```
; printer driver file = /usr/local/samba/printers.def
```

```
# It should not be necessary to spell out the print system type unless  
# yours is non-standard. Currently supported print systems include:
```

```
# bsd, sysv, plp, lprng, aix, hpux, qnx
```

```
printing = lprng
```

```
# Uncomment this if you want a guest account, you must add this to  
# /etc/passwd otherwise the user "nobody" is used
```

```
; guest account = pguest
```

```
# this tells Samba to use a separate log file for each machine that connects
```

```
log file = /var/log/samba/log.%m
```

```

# Put a capping on the size of the log files (in Kb).
    max log size = 50

# Security mode. Most people will want user level security. See
# security_level.txt for details.
    security = domain

# Use password server option only with security = server
;    password server = SAMBASYS

# Password Level allows matching of _n_ characters of the password for
# all combinations of upper and lower case.
    password level = 8
    username level = 8

# You may wish to use password encryption. Please read
# Encryption.txt, Win95.txt and WinNT.txt in the Samba documentation
# Do not enable this option unless you have read those documents

    encrypt passwords = yes
    smb passwd file = /etc/smbpasswd

# The following are needed to allow password changing from Windows to
# update the Linux system password also.
# NOTE: Use these with 'encrypt passwords' and 'smb passwd file' above.
# NOTE2: You do NOT need these to allow workstations to change only
# the encrypted SMB passwords. They allow the Unix password
# to be kept in sync with the SMB password.
;    unix password sync = Yes
;    passwd program = /usr/bin/passwd %u
;    passwd chat = *New*UNIX*password*      %n\n
*ReType*new*UNIX*password*                  %n\n
*passwd:*all*authentication*tokens*updated*successfully*

# Unix users can map to different SMB User names
;    username map = /etc/smbusers

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name

```


of the machine that is connecting

; include = /etc/smb.conf.%m

Most people will find that this option gives better performance.

See speed.txt and the manual pages for details

socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

Configure Samba to use multiple interfaces

If you have multiple network interfaces then you must list them here

See the man page for details.

; interfaces = 192.168.12.2/24 192.168.13.2/24

Configure remote browse list synchronisation here

Request announcement to, or browse list sync from a specific host or

from / to a whole subnet (see below)

; remote browse sync = 192.168.3.25 192.168.5.255

Cause this host to announce it to local subnets here

remote announce = 192.168.1.255

Browser Control Options:

set local master to no if you don't want Samba to become a master

browser on your network. Otherwise the normal election rules apply

local master = no

OS Level determines the precedence of this server in master browser

elections. The default value should be reasonable

os level = 0

Domain Master specifies Samba to be the Domain Master Browser. This

allows Samba to collate browse lists between subnets. Don't use this

if you already have a Windows NT domain controller doing this job

domain master = no

Preferred Master causes Samba to force a local browser election on startup

```
# and gives it a slightly higher chance of winning the election
```

```
    preferred master = no
```

```
# Use only if you have an NT server on your network that has been  
# configured at install time to be a primary domain controller.
```

```
;    domain controller = sambaserver  
;    <NT-Domain-Controller-SMBName>
```

```
# Enable this if you want Samba to be a domain logon server for  
# Windows95 workstations  
    domain logons = yes
```

```
# if you enable domain logons then you may want a per-machine or  
# per user logon script
```

```
# run a specific logon batch file per workstation (machine)  
;    logon script = %m.bat
```

```
# run a specific logon batch file per username  
;    logon script = %U.bat
```

```
# Where to store roving profiles (only for Win95 and WinNT)  
#    %L substitutes for this servers netbios name, %U is username  
#    You must uncomment the [Profiles] share below
```

```
    logon path = \\%L\Profiles\%U
```

```
# All NetBIOS names must be resolved to IP Addresses  
# 'Name Resolve Order' allows the named resolution mechanism to be  
# specified the default order is "host lmhosts wins bcast". "host" means use  
# the Unix system gethostbyname() function call that will use either  
# /etc/hosts OR DNS or NIS depending on the settings of /etc/host.config,  
# /etc/nsswitch.conf and the /etc/resolv.conf file.  
# "host" therefore is system configuration dependant. This parameter is most  
# often of use to prevent DNS lookups in order to resolve NetBIOS names to  
# IP Addresses. Use with care!
```

```
# The example below excludes use of name resolution for machines that are
```

```
# NOT on the local network segment
# - OR - are not deliberately to be known via lmhosts or via WINS.

;    name resolve order = wins lmhosts bcact

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable it's
#WINS Server

    wins support = yes

# Tells the NMBD components of Samba to be a WINS Client
# Note: Samba can be either a WINS Server, or a WINS Client, but NOT
#both
;    wins server = w.x.y.z

# WINS Proxy - Tells Samba to answer name resolution queries on
# behalf of a non-WINS capable client, for this to work there must be
# at least one    WINS Server on the network. The default is NO.

;    wins proxy = yes

# DNS Proxy - tells Samba whether or not to try to resolve NetBIOS names
# via DNS nslookups. The built-in default for versions 1.9.17 is yes,
# this has been changed in version 1.9.18 to no.

    dns proxy = no

# Case Preservation can be handy - system default is _no_
# NOTE: These can be set on a per share basis

;    preserve case = no
;    short preserve case = no

# Default case is normally upper case for all DOS files
;    default case = lower

# Be very careful with case sensitivity - it can break things!
;    case sensitive = no
```

```
# ===== Share Definitions =====
```

```
[homes]
```

```
comment = Home Directories
path = %H
valid users = %S
browseable = no
writeable = yes
create mode = 0700
directory mode = 0700
available = yes
public = no
only user = no
```

```
# Un-comment the following and create the netlogon directory for Domain
#Logons
```

```
[netlogon]
```

```
comment = Network Logon Service
path = /home/netlogon
guest ok = yes
writeable = no
share modes = no
available = yes
```

```
# Un-comment the following to provide a specific roving profile share
# the default is to use the user's home directory
```

```
[Profiles]
```

```
path = /home/profiles
browseable = no
guest ok = yes
```

```
# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
```

```
[printers]
```

```
comment = Printers
path = /var/spool/lpd/lp
browseable = yes
printable = yes
```

```
public = yes  
writeable = no  
create mask = 0700
```

CONFIGURATIONS

Global Settings: [global]

Workgroup

```
workgroup = STUDENT
```

It specifies in which Windows NT workgroup the Samba server will participate. It is equivalent to Windows NT domain or workgroup name.

NetBIOS Name

```
netbios name = SAMBASYS
```

This is the name the Samba server is known by on the network. This parameter has the same meaning as a Windows NT computer name. If you do not specify, it defaults to the server's hostname.

Server String

```
server string = Samba Server
```

This is the description string of the Samba server. It has the same role as the Windows NT description field.

Hosts Allowed

```
hosts allow = 192.168.1.
```

It allows you to restrict connections to machines, which are on your local network.

Loading Printers

```
printcap name = /etc/printcap
load printers = yes
printing = lprng
```

These commands are used if we want to load printers automatically instead of defining them individually. In **printing** command we have different types of supported formats for printing. **lprng** stands for **L**ine **P**Rinters **N**ext **G**eneration. By default this style is used by Samba.

Log files

```
log file = /var/log/samba/log.%m
max log size = 50
```

This tells Samba to use a separate log file for each machine that connects. It makes entries for each user who logged on the server and the time user was logged in with Samba server.

Security modes

```
security = user
```

This parameter has four possible values:

1. **Share:** For this security mode, clients need to supply only the password for the resource. This mode of security is the default for the Windows 95 file and print servers.
2. **User:** User & Password validation is done on the server that is offering the resource. This mode is the most widely used.
3. **Server:** The User & Password validation is done on the specified authentication server. This server can be a Windows NT server or another Samba server.
4. **Domain:** This security level is basically the same as server security, with the exception that the Samba server becomes a member of a Windows NT domain. In this case, the Samba server can also participate in such things as trust relationships.

If security = server is used then we must also use this command

```
password server = <NT-Server-Name>
```

This specifies that from which server it get the list of passwords.

Password Levels

```
password level = 8  
username level = 8
```

It allows matching of n characters of the password for all combinations of upper and lower case. As in above case, **8** characters are matched. Increasing the levels will increase the security of the system.

Password encryption

```
encrypt passwords = yes
```

This parameter is used if you want to enable Samba to use the encrypted password protocol.

```
smb passwd file = /etc/smbpasswd
```

This parameter tells Samba where the encrypted password file is. Before you start the Samba server for the first time, you need to create a Samba encrypted passwords file.

To add all the current users to the Samba server, run the **mksmbpasswd.sh** utility as:

```
cat /etc/passwd | /usr/bin/mksmbpasswd.sh > /etc/smbpasswd
```

Now all of the Linux users become Samba users and their username are added in the Samba password file. Although the users have been added in the file but still you need to update the passwords individually by using this command:

```
/usr/bin/smbpasswd -U username
```

After running the **mksmbpasswd** utility, all Linux users are become Samba user and password are assigned to them individually. Now if we want to add

new Linux user than we have to use following command to make him/her a Samba user too.

```
/usr/bin/smbpasswd -a username password
```

Sockets Options

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

This option helps in providing better performance when connection is created between the two machines.

Remote Announcement

```
remote announce = 192.168.1.255
```

This option is used if you want to announce that your server is present on the network.

Browser Control Options

LOCAL MASTER

```
local master = no
```

If you used as Local Master then set this option to yes. Otherwise, set local master to no on your network.

OS LEVEL

```
os level = 0
```

It determines the precedence of this server in master browser elections. The default value should be at some reasonable value.

DOMAIN MASTER

```
domain master = no
```


It specifies Samba to be the Domain Master Browser. This allows Samba to collate browse lists between subnets. Don't use this if you already have a Windows NT domain controller doing this job.

PREFERRED MASTER

preferred master = no

It causes Samba to force a local browser election on start-up and gives it a slightly higher chance of winning the election.

Domain Logon Server

domain logons = no

This option is used if you want Samba to be a domain logon server for Windows95 workstations. With this feature it is possible for user to logon to network from NT machines.

If domain logons are yes then following options are also used for logon of different user. These are scripts that may be used as per machine or per user.

Run a specific logon batch file per workstation (machine)

; logon script = %m.bat

Run a specific logon batch file per username

; logon script = %U.bat

Name Resolve Order

name resolve order = wins lmhosts bcst

With this parameter you specify how the Samba server will resolve NetBIOS names into IP addresses. The preferred value is **wins lmhosts bcst**.

Windows Internet Name Serving (WINS) Support

wins support = yes

It tells the NMBD component of Samba to enable its WINS Server.

DNS Proxy

```
dns proxy = no
```

It tells Samba whether or not to try to resolve NetBIOS names via DNS nslookups.

Share Definitions:**Home Directories [homes]:**

```
comment = Home Directories
path = %H
valid users = %S
browseable = no
writable = yes
create mode = 0700
directory mode = 0700
available = yes
public = no
only user = no
```

It instructs Samba to dynamically create a share using the username sent by the client. If value has not been set for the path samba uses the user home directory path defined in the system's `/etc/passwd` file.

When a client requests a connection to a share, existing shares are scanned. If a match is found, that share is used. If no match is found, the requested share is treated as the username and validated by security. If the name exists and the password is correct, a share with that name is created by cloning the [homes] section. Home share definitions use the same parameters as normal shares.

Profiles: [profiles]

```
path = /home/profiles
browseable = no
guest ok = yes
```

The location where Windows stores the roaming profile on the network is set by the logon path parameter in smb.conf. The default value for this is a subdirectory in the user's home directory named profile.

This will allow you to easily set up a new share. Each share definition starts with the share name in brackets “[]”. Following the share name you specify the share parameters and values.

Share Permissions: []

Although you can control the share permissions with share parameters, Linux permissions are applied before share permissions. Make sure the Linux permissions let the users access the share directory in the Linux environment. When a user creates a new file on the shared directory, the default create mask used is 0744. For directory creation, the default create mask is 0755.

Printing: [printers]

```
comment = All Printers
path = /var/spool/samba
browseable = no
guest ok = no
writable = no
printable = yes
create mask = 0700
```

A Samba server uses the same procedure for [printer] shares as for [home] shares. The share definitions and user names are tested against the requested share name. If a match is found in the [printers] share section, a share with that name is cloned with the name of the requested service.

The [printers] section is just like the other share definitions. When a user prints, Samba copies the data into the spool directory, after which it is handled by the local printing system. The only big difference between a printer share and other share definitions is that the printable parameter is set to “yes”. This means that a user can write a spool file to the directory

specified under the share definition. If the share is printable, then it is also write-able by default.

SUMMARY

The Samba is free software, which is used to communicate between two different environments of Windows and Linux. It is usually used to reduce the licensing cost of the highly priced products e.g. we can replace an NT print server with Samba Print Server. Samba uses Server Message Block (SMB) protocol, which is a connection-oriented service that uses the NetBIOS session service. SMB helps in the successful connection of Windows and Linux machine.

Samba provides many tools for automating activities on the server sides of SMB connections. Samba can effectively act as a File and Printer server for PC in a group environment like LAN. Using Samba, the resources of Linux and Unix machines can be used on Windows NT and vice versa.