Linux WWW HOWTO

Table of Contents

Table of Contents

Table of Contents

Linux WWW HOWTO

by Mr. Poet, `poet@linuxports.com`

v0.85, 21 August 1999

This document contains information about setting up WWW services under Linux (both server and client). It tries not to be a in detail manual but an overview and a good pointer to further information.

6.Setting up WWW server systems

7.Apache

8.Web Server Add−ons

9.Intranet Section

10.Install the HTTP server

11.Testing HTTPd

12.Connecting to the Linux Server

13.Setting up the Intranet

14.Accessing the Web

15.More things to do

16.Credits and Legalities

17.For further reading

1.Introduction

Many people are trying Linux because they are looking for a really good *Internet capable* operating system. Also, there are institutes, universities, non−profits, and small businesses which want to set up Internet sites on a small budget. This is where the WWW−HOWTO comes in. This document explains how to set up clients and servers for the largest part of the Internet − *The World Wide Web*.

All prices in this document are stated in US dollars. This document assumes you are running Linux on an Intel platform. Instructions and product availability my vary from platform to platform. There are many links for downloading software in this document. Whenever possible use a mirror site for faster downloading and to keep the load down on the main server.

1.1 Copyright

1.2 Feedback

Any feedback is welcome. I do not claim to be an expert. Some of this information was taken from badly written web sites; there are bound to be errors and omissions. But make sure you have the latest version before you send corrections; It may be fixed in the next version (see the next section for where to get the latest version). Send feedback to poet@linuxports.com.

1.3 New versions of this Document

New versions of this document are always available at the LinuxPorts Website.

2.Setting up WWW client software (Antiquated)

> Please note that the following section is very outdated and was last updated in 1997. This section is being kept here for historical purposes only. All major distributions of Linux come with Netscape and Lynx. These are the two most popular browsers for the Linux Operating System. If you do not have these browsers installed currently please go to their respective sections for download information.

The following chapter is dedicated to the setting up web browsers. Please feel free to contact me, if your favorite web browser is not mentioned here. In this version of the document only a few of the browsers have there own section, but I tried to include all of them (all I could find) in the overview section. In the future those browsers that deserve there own section will have it.

The overview section is designed to help you decide which browser to use, and give you basic information on each browser. The detail section is designed to help you install, configure, and maintain the browser.

However I use Lynx when I don't feel like firing up the X−windows/Netscape monster.

2.1 Overview

Navigator/Communicator

Netscape Navigator is the only Linux browser mentioned here, which is capable of advanced HTML features. Some of these features are frames, Java, Javascript, automatic update, and layers. It also has news and mail capability. But it is a resource hog; it takes up lots of CPU time and memory. It also sets up a separate cache for each user wasting disk space.
Netscape is currently an OpenSource product and can be downloaded from ftp.netscape.com=>. <tag><ref id= name="Lynx"> Lynx is the one of the smallest web browsers. It is the king of text based browsers. It's free and the source code is available under the GNU public license. It's text based, but it has many special features. Lynx now supports tables, color (via curses) and frames.
Note on frame support for lynx:
The frame support for lynx is limited, it will notice the frames and show the title of the frames for your to select as hot links. Since, frame titles are usually very undescriptive for coding simplicity this can be confusing.

Kfm

Kfm is part of the K Desktop Environment (KDE). KDE is a system that runs on top of X−windows. It gives you many features like drag an drop, sounds, a trashcan and a unified look and feel. Kfm is the K File Manager, but it is also a web browser. It is very usable as a web browser and it supports frames, tables, ftp downloads, looking into tar files, and more. The current release of KDE is 1.1.1 with 1.1.2 very near by. Kfm can be used without KDE, but you still need the librarys that come with KDE. For more information about KDE and Kfm visit the KDE website at http://www.kde.org.

Emacs

Emacs is the one program that does everything. It is a word processor, news reader, mail reader, and web browser. It has a steep learning curve at first, because you have to learn what all the keys do. The X−windows version is easier to use, because most of the functions are on menus. Another drawback is that it's mostly text based. (It can display graphics if you are running it under X−windows). It is also free, and the source code is available under the GNU public license.

NCSA Mosaic

Mosaic is an X−windows browser developed by the National Center for Supercomputing Applications (NCSA) at the University of Illinois. NCSA spent four years on the project and has now moved on to other things. Again, Mosaic is no longer supported. However since the source is free for non−commercial use it might make an interesting project for someone who wants to develop a new browser.

Amaya

Amaya is the X−windows concept browser for the W3C for HTML 3.2. Therefore it supports all the HTML 3.2 standards. It also supports some of the features of HTML 4.0. It supports tables, forms, client side image maps, put publishing, gifs, jpegs, and png graphics. It is both a browser and authoring tool. The latest public release is 1.0 beta. Version 1.1 beta is in internal testing and is due out soon. For more information visit the Amaya web site at http://www.w3.org/Amaya/. It can be downloaded from ftp://ftp.w3.org/pub/Amaya−LINUX−ELF−1.0b.tar.gz.

Qweb

Qweb is yet another basic X−windows browser. It supports tables, forms, and server site image maps. The latest version is 1.3. For more information visit the Qweb website at http://sunsite.auc.dk/qweb/ The source is available from http://sunsite.auc.dk/qweb/qweb−1.3.tar.gz The binaries are available in a Red Hat RPM from http://sunsite.auc.dk/qweb/qweb−1.3−1.i386.rpm

It is the reccomendation of this author that users of web browsers use either Netscape 4.x, Lynx or Netscape 5.xAlpha. They are the only one currently available for Linux that support most features. Personally I suggest Netscape5.xAlpha, even though it is Alpha Software it is quite stable and frankly implements the standards better than the 4.x versions do.

3.Lynx

Lynx is one of the smaller (around 600 K executable) and faster web browsers available. It does not eat up much bandwidth nor system resources as it only deals with text displays. It can display on any console, terminal or xterm. You will not need an *X Windows system* or additional system memory to run this little browser.

3.1 Where to get

Most distributions have Lynx in them. Therefore I will not bore you with the details of compiling and installing Lynx.

The latest version is 2.8.2 and can be retrieved from http://www.slcc.edu/lynx/fote/ or from almost any friendly Linux FTP server like ftp://sunsite.unc.edu under /pub/Linux/apps/www/broswers/ or mirror site.

For more information on Lynx try these locations:

Lynx Links

> http://www.crl.com/~subir/lynx.html

Lynx Pages

> http://lynx.browser.org

Lynx Help Pages

> http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html (the same pages you get from lynx −−help and typing ? in lynx)

Note: The Lynx help pages have recently moved. If you have an older version of Lynx, you will need to change your lynx.cfg (in /usr/lib) to point to the new address(above).

I think the most special feature of Lynx against all other web browsers is the capability for batch mode retrieval. One can write a shell script which retrieves a document, file or anything like that via *http*, *FTP*, *gopher*, *WAIS*, *NNTP* or *file://* − url's and save it to disk. Furthermore, one can fill in data into HTML forms in batch mode by simply redirecting the standard input and using the −*post_data* option.

For more special features of Lynx just look at the help files and the man pages. If you use a special feature of Lynx that you would like to see added to this document, let me know.

4.Emacs−W3

There are several different flavors of Emacs. The two most popular are GNU Emacs and XEmacs. GNU Emacs is put out by the Free Software Foundation, and is the original Emacs. It is mainly geared toward text based terminals, but it does run in X−Windows. XEmacs (formerly Lucid Emacs) is a version that only runs on X−Windows. It has many special features that are X−Windows related (better menus etc).

4.1 Where to get

Most distributions include GNU Emacs.

The most recent GNU emacs is 19.34. It doesn't seem to have a web site. The FTP site is at ftp://ftp.gnu.ai.mit.edu/pub/gnu/.

The latest version of XEmacs is 20.2. The XEmacs FTP site is at ftp://ftp.xemacs.org/pub/xemacs. For more information about XEmacs goto see its web page at http://www.xemacs.org.

Both are available from the Linux archives at ftp://sunsite.unc.edu under /pub/Linux/apps/editors/emacs/

If you got GNU Emacs or XEmacs installed, you probably got the W3 browser running to.

The Emacs W3 mode is a nearly fully featured web browser system written in the Emacs Lisp system. It mostly deals with text, but can display graphics, too − at least − if you run the emacs under the X Window system.

To get XEmacs in to W3 mode, goto the apps menu and select browse the web.

I don't use Emacs, so if someone will explain how to get it into the W3 mode I'll add it to this document. Most of this information was from the original author. If any information is incorrect, please let me know. Also let me know if you think anything else should be added about Emacs.

5.Netscape Navigator/Communicator

5.1 Different versions and options.

Netscape Navigator is the King of WWW browsers. Netscape Navigator can do almost everything. But on the other hand, it is one of the most memory hungry and resource eating program I've ever seen.

There are 3 different versions of the program:

Netscape Navigator includes the web browser, netcaster (push client) and a basic mail program.

Netscape Communicator includes the web browser, a web editor, an advanced mail program, a news reader, netcaster (push client), and a group conference utility.

Netscape Communicator Pro includes everything Communicator has plus a group calendar, IBM terminal emulation, and remote administration features (administrators can update thousands of copies of Netscape from their desk).

In addition to the three versions there are two other options you must pick.

The first is full install or base install. The full install includes everything. The base install includes enough to get you started. You can download the additional components as you need them (such as multimedia support and netcaster). These components can be installed by the Netscape smart update utility (after installing goto help−>software updates). At this time the full install is not available for Linux.

The second option is import or export. If you are from the US are Canada you have the option of selecting the import version. This gives you the stronger 128 bit encryption for secure transactions (SSL). The export version only has 40 bit encryption, and is the only version allowed outside the US and Canada.

The latest version of the Netscape Navigator/Communicator/Communicator Pro is 4.6.1. There are two different versions for Linux. One is for the old 1.2 series kernels and one for the new 2.x kernels. If you don't have at least a 2.0 kernel I suggest you upgrade; there are many improvements in the new kernel.

Beta versions are also available. If you try a beta version, they usually expire in a month or so!

You can also try the 5.0 alpha at the Mozilla project located at www.mozilla.org. I suggest this highly, the new Netscape 5 is the best browser I have seen in a long time. It is extremely flexible and mostly stable even for an Alpha release.

5.2 Where to get

The best way to get Netscape software is to go through their web site at http://www.netscape.com/download/. They have menu's to guide you through the selection. When it ask for the Linux version, it is referring to the kernel (most people should be using 2.0 by now). If your not sure which version kernel you have run 'cat /proc/version'. Going through the web site is the only way to get the import versions.

If you want an export version you can download them directly from the Netscape FTP servers. The FTP servers are also more up to date. For example when I first wrote this the web interface did not have the non−beta 4.03 for Linux yet, but it was on the FTP site. Here are the links to the export Linux 2.0 and 2.2 versions:

Netscape Navigator 4.6.1 is at
ftp://ftp.netscape.com/pub/communicator/4.6/shipping/english/unix/linux20/navigator_standalone/navigator−v403−export.x86−unknown−linux2.0.tar.gz

Netscape Communicator 4.6.1 for Linux 2.0 (kernel) is at
ftp://ftp.netscape.com/pub/communicator/4.6/shipping/english/unix/linux20/base_install/communicator−v403−export.x86−unknown−linux2.0.tar.gz

These url's will change as new versions come out. If these links break you can find them by fishing around at the FTP site ftp://ftp.netscape.com/pub/communicator/.

These servers are heavily loaded at times. Its best to wait for off peak hours or select a mirror site. Be prepared to wait, these archives are large. Navigator is almost 8megs, and Communicator base install is 10megs.

5.3 Installing

This section explains how to install version 4 of Netscape Navigator, Communicator, and Communicator Pro.

First unpack the archive to a temporary directory. Then run the `ns−install` script (type `./ns−install`). Then make a symbolic link from the `/usr/local/netscape/netscape` binary to `/usr/local/bin/netscape` (type `ln −s /usr/local/netscape/netscape /usr/local/bin/netscape`). Finally set the system wide environment variable `$MOZILLA_HOME` to `/usr/local/netscape` so Netscape can find its files. If you are using bash for your shell edit your `/etc/profile` and add the lines:

```
MOZILLA_HOME="/usr/local/netscape"
export MOZILLA_HOME
```

After you have it installed the software can automatically update itself with smart update. Just run Netscape as root and goto help−>software updates. If you only got the base install, you can also install the Netscape components from there.

Note: This will not remove any old versions of Netscape, you must manually remove them by deleting the Netscape binary and Java class file (for version 3).

6.Setting up WWW server systems

This section contains information on different http server software packages and additional server side tools like script languages for CGI programs etc. There are several dozen web servers, I only covered those that are fully functional. As some of these are commercial products, I have no way of trying them. Most of the information in the overview section was pieced together from various web sites. If there is any incorrect or missing information please let me know.

For a technical description on the http mechanism, take a look at the RFC documents mentioned in the chapter "For further reading" of this HOWTO.

I prefer to use the Apache server. It has almost all the features you would ever need and its free! I will admit that this section is heavily biased toward Apache. I decided to concentrate my efforts on the Apache section rather than spread it out over all the web servers. I may cover other web servers in the future.

6.1 Overview

Cern httpd

This was the first web server. It was developed by the European Laboratory for Particle Physics (CERN). CERN httpd is no longer supported. The CERN httpd server is reported to have some ugly bugs, to be quite slow and resource hungry. The latest version is 3.0. For more information visit the CERN httpd home page at http://www.w3.org/Daemon/Status.html. It is available for download at ftp://sunsite.unc.edu/pub/Linux/apps/www/servers/httpd−3.0.term.tpz (no it is not a typo, the extension is actually .tpz on the site; probably should be .tgz)

NCSA HTTPd

The NCSA HTTPd server is the father to Apache (The development split into two different servers). Therefore the setup files are very similar. NCSA HTTPd is free and the source code is available. This server not covered in this document, although reading the Apache section may give you some help. The NCSA server was once popular, but most people are replacing it with Apache. Apache is a drop in replacement for the NCSA server(same configuration files), and it fixes several shortcomings of the NCSA server. NCSA HTTPd accounts for 4.9% (and falling) of all web servers. (source September 1997 Netcraft survey). The latest version is 1.5.2a. For more information see the NCSA website at http://hoohoo.ncsa.uiuc.edu.

Apache

Apache is the king of all web servers. Apache and its source code is free. Apache is modular, therefore it is easy to add features. Apache is very flexible and has many, many features. Apache and its derivatives makes up 55% of all web domains. There are over 3,928,112 Apache servers in operation (source August 1999 Netcraft survey).
The official Apache is missing SSL, but there are two derivatives that fill the gap. Stronghold is a commercial product that is based on Apache. It retails for $995; an economy version is available for $495 (based on an old version of Apache). Stronghold is the number two secure server behind Netscape (source C2 net and Netcraft survey). For more information visit the Stronghold website at http://www.c2.net/products/stronghold/. It was developed outside the US, so it is available with 128 bit SSL everywhere.
Apache−SSL is a free implementation of SSL, but it is not for commercial use in the US (RSA has US patents on SSL technology). It can be used for non−commercial use in the US

if you link with the free RSAREF library. For more information see the website at http://www.algroup.co.uk/Apache−SSL/.

Netscape Fast Track Server

Fast Track was developed by Netscape, but the Linux version is put out by Caldera. The Caldera site lists it as Fast Track for OpenLinux. I'm not sure if it only runs on Caldera OpenLinux or if any Linux distribution will do (E−mail me if you have the answer). Netscape servers account for 6.11% (and RISING!) of all web servers (source August 1999 http://www.netcraft.com/survey/). The server sells for $295. It is also included with the Caldera OpenLinux Standard distribution which sells for $399 ($199.50 educational). The web pages tell of a nice administration interface and a quick 10 minute setup. The server has support for 40−bit SSL. To get the full 128−bit SSL you need Netscape Enterprise Server. Unfortunately that is not available for Linux :(The latest version available for Linux is 2.0 (Version 3 is in beta, but its not available for Linux yet). To buy a copy goto the Caldera web site at http://www.caldera.com/products/netscape/netscape.html For more information goto the Fast Track page at http://www.netscape.com/comprod/server_central/product/fast_track/

WN

WN has many features that make it attractive. First it is smaller than the CERN, NCSA HTTPd, an Apache servers. It also has many built−in features that would require CGI's. For example site searches, enhanced server side includes. It can also decompress/compress files on the fly with its filter feature. It also has the ability to retrieve only part of a file with its ranges feature. It is released under the GNU public license. The current version is 2.08. For more information see the WN website at http://hopf.math.nwu.edu/.

AOLserver

AOLserver is made by America Online. I'll admit that I was surprised by the features of a web server coming from AOL. In addition to the standard features it supports database connectivity. Pages can query a database by Structured Query Language (SQL) commands. The database is access through Open Database Connectivity (ODBC). It also has built−in search engine and TCL scripting. If that is not enough you can add your own modules through the c Application Programming Interface (API). I almost forgot to mention support for 40 bit SSL. And you get all this for free! For more information visit the AOLserver site at http://www.aolserver.com/server/

CL−HTTP

CL−HTTP stands for Common Lisp Hypermedia Server. If you are a Lisp programmer this server is for you. You can write your CGI scripts in Lisp. It has a web based setup function. It also supports all the standard server features. CL−HTTP is free and the source code is available. For more information visit the CL−HTTP website at http://www.ai.mit.edu/projects/iiip/doc/cl−http/home−page.html

If you have a commercial purpose (company web site, or ISP), I would strongly recommend that you use Apache. I've also heard that the Netscape Server is easy to setup. If you have an internal use you can be a bit more flexible. But unless one of them has a feature that you just have to use,

This is only a partial listing of all the servers available. For a more complete list visit Netcraft at http://www.netcraft.com/survey/servers.html or Web Compare at http://webcompare.internet.com.

7.Apache

The current version of Apache is 1.3.9. The main Apache site is at http://www.apache.org/. Another good source of information is Apacheweek at http://www.apacheweek.com/. The Apache documentation is ok, so I'm not going to go into detail in setting up apache. The documentation is on the website and is included with the source (in HTML format). There are also text files included with the source, but the HTML version is better. The documentation should get a whole lot better once the Apache Documentation Project gets under way. Right now most of the documents are written by the developers. Not to discredit the developers, but they are a little hard to understand if you don't know the terminology.

7.1 Where to get

Apache is included in the Red Hat, Slackware, and OpenLinux distributions. Although they may not be the latest version, they are very reliable binaries. The bad news is you will have to live with their directory choices (which are totally different from each other and the Apache defaults).

The source is available from the Apache web site at http://www.apache.org/dist/ Binaries are are also available at apache at the same place. You can also get binaries from sunsite at ftp://sunsite.unc.edu/pub/Linux/apps/www/servers/. And for those of us running Red Hat the latest binary RPM file can usually be found in the contrib directory at ftp://ftp.redhat.com/pub/contrib/i386/

If your server is going to be used for commercial purposes, it is highly recommended that you get the source from the Apache website and compile it yourself. The other option is to use a binary that comes with a major distribution. For example Slackware, Red Hat, or OpenLinux distributions. The main reason for this is security. An unknown binary could have a back door for hackers, or an unstable patch that could crash your system. This also gives you more control over what modules are compiled in, and allows you to set the default directories. It's not that difficult to compile Apache, and besides you not a real Linux user until you compile your own programs ;)

7.2 Compiling and Installing

First untar the archive to a temporary directory. Next change to the src directory. Then edit the Configuration file if you want to include any special modules. The most commonly used modules are already included. There is no need to change the rules or makefile stuff for Linux. Next run the Configure shell script (`./Configure`). Make sure it says Linux platform and gcc as the compiler. Next you may want to edit the httpd.h file to change the default directories. The server home (where the config files are kept) default is `/usr/local/etc/httpd/`, but you may want to change it to just `/etc/httpd/`. And the server root (where the HTML pages are served from) default is `/usr/local/etc/httpd/htdocs/`, but I like the directory `/home/httpd/html` (the Red Hat default for Apache). If you are going to be using su−exec (see special features below) you may want to change that directory too. The server root can also be changed from the config files too. But it is also good to compile it in, just encase Apache can't find or read the config file. Everything else should be changed from the config files. Finally run make to compile Apache.

If you run in to problems with include files missing, check the following things. Make sure you have the kernel headers (include files) installed for your kernel version. Also make sure you have these symbolic links in place:

```
/usr/include/linux should be a link to /usr/src/linux/include/linux
/usr/include/asm should be a link to /usr/src/linux/include/asm
/usr/src/linux should be a link to the Linux source directory (ex.linux-2.0.30)
```

Links can be made with `ln −s`, it works just like the cp command except it makes a link (`ln −s source-dir destination−link`)

When make is finished there should be an executable named httpd in the directory. This needs to be moved in to a bin directory. `/usr/sbin` or `/usr/local/sbin` would be good choices.

Copy the conf, logs, and icons sub−directories from the source to the server home directory. Next rename 3 of the files files in the conf sub−directory to get rid of the `−dist` extension (ex. `httpd.conf−dist` becomes `httpd.conf`)

There are also several support programs that are included with Apache. They are in the `support` directory and must be compiled and installed separately. Most of them can be make by using the makefile in that directory (which is made when you run the main `Configure` script). You don't need any of them to run Apache, but some of them make the administrators job easier.

7.3 Configuring

Now you should have four files in your `conf` sub−directory (under your server home directory). The `httpd.conf` sets up the server daemon (port number, user, etc). The `srm.conf` sets the root document tree, special handlers, etc. The `access.conf` sets the base case for access. Finally `mime.types` tells the server what mime type to send to the browser for each extension.

The configuration files are pretty much self−documented (plenty of comments), as long as you understand the lingo. You should read through them thoroughly before putting your server to work. Each configuration item is covered in the Apache documentation.

The `mime.types` file is not really a configuration file. It is used by the server to translate file extensions into mime−types to send to the browser. Most of the common mime−types are already in the file. Most people should not need to edit this file. As time goes on, more mime types will be added to support new programs. The best thing to do is get a new mime−types file (and maybe a new version of the server) at that time.

Always remember when you change the configuration files you need to restart Apache or send it the SIGHUP signal with `kill` for the changes to take effect. Make sure you send the signal to the parent process and not any of the child processes. The parent usually has the lowest process id number. The process id of the parent is also in the `httpd.pid` file in the log directory. If you accidently send it to one of the child processes the child will die and the parent will restart it.

I will not be walking you through the steps of configuring Apache. Instead I will deal with specific issues, choices to be made, and special features.

I highly recommend that all users read through the security tips in the Apache documentation. It is also available from the Apache website at http://www.apache.org/docs/mics/security_tips.html.

7.4 Hosting virtual websites

Virtual Hosting is when one computer has more than one domain name. The old way was to have each virtual host have its own IP address. The new way uses only one IP address, but it doesn't work correctly with browsers that don't support HTTP 1.1.

My recommendation for businesses is to go with the IP based virtual hosting until most people have browsers that support HTTP 1.1 (give it a year or two). This also gives you a more complete illusion of virtual hosting. While both methods can give you virtual mail capabilities (can someone confirm this?), only IP based virtual hosting can also give you virtual FTP as well.

If it is for a club or personal page, you may want to consider shared IP virtual hosting. It should be cheaper than IP based hosting and you will be saving precious IP addresses.

You can also mix and match IP and shared IP virtual hosts on the same server. For more information on virtual hosting visit Apacheweek at http://www.apacheweek.com/features/vhost.

IP based virtual hosting

In this method each virtual host has its own IP address. By determining the IP address that the request was sent to, Apache and other programs can tell what domain to serve. This is an incredible waste of IP space. Take for example the servers where my virtual domain is kept. They have over 35,000 virtual accounts, that means 35,000 IP addresses. Yet I believe at last count they had less than 50 servers running.

Setting this up is a two part process. The first is getting Linux setup to accept more than one IP address. The second is setting up apache to serve the virtual hosts.

The first step in setting up Linux to accept multiple IP addresses is to make a new kernel. This works best with a 2.0 series kernel (or higher). You need to include IP networking and IP aliasing support. If you need help with compiling the kernel see the kernel howto.

Next you need to setup each interface at boot. If you are using the Red Hat Distribution then this can be done from the control panel. Start X−windows as root, you should see a control panel. Then double click on network configuration. Next goto the interfaces panel and select your network card. Then click alias at the bottom of the screen. Fill in the information and click done. This will need to be done for each virtual host/IP address.

If you are using other distributions you may have to do it manually. You can just put the commands in the `rc.local` file in `/etc/rc.d` (really they should go in with the networking stuff). You need to have a `ifconfig` and `route` command for each device. The aliased addresses are given a sub device of the main one. For example eth0 would have aliases eth0:0, eth0:1, eth0:2, etc. Here is an example of configuring a aliased device:

```
ifconfig eth0:0 192.168.1.57
route add -host 192.168.1.57 dev eth0:0
```

You can also add a broadcast address and a netmask to the ifconfig command. If you have alot of aliases you may want to make a for loop to make it easier. For more information see the IP alias mini howto.

Then you need to setup your domain name server (DNS) to serve these new domains. And if you don't already own the domain names, you need to contact the Internic to register the domain names. See the DNS−howto for information on setting up your DNS.

Finally you need to setup Apache to server the virtual domain correctly. This is in the `httpd.conf` configuration file near the end. They give you an example to go by. All commands specific to that virtual host are put in between the `virtualhost` directive tags. You can put almost any command in there. Usually you set up a different document root, script directory, and log files. You can have almost unlimited number of virtual hosts by adding more `virtualhost` directive tags.

In rare cases you may need to run separate servers if a directive is needed for a virtual host, but is not allowed in the virtual host tags. This is done using the bindaddress directive. Each server will have a different name and setup files. Each server only responds to one IP address, specified by the bindaddress directive. This is an incredible waste of system resources.

Shared IP virtual hosting

This is a new way to do virtual hosting. It uses a single IP address, thus conserving IP addresses for real machines (not virtual ones). In the same example used above those 30,000 virtual hosts would only take 50 IP addresses (one for each machine). This is done by using the new HTTP 1.1 protocol. The browser tells the server which site it wants when it sends the request. The problem is browsers that don't support HTTP 1.1 will get the servers main page, which could be setup to provide a menu of virtual hosts available. That ruins the whole illusion of virtual hosting. The illusion that you have your own server.

The setup is much simpler than the IP based virtual hosting. You still need to get your domain from the Internic and setup your DNS. This time the DNS points to the same IP address as the original domain. Then Apache is setup the same as before. Since you are using the same IP address in the virtualhost tags, it knows you want Shared IP virtual hosting.

There are several work arounds for older browsers. I'll explain the best one. First you need to make your main pages a virtual host (either IP based or shared IP). This frees up the main page for a link list to all your virtual hosts. Next you need to make a back door for the old browsers to get in. This is done using the `ServerPath` directive for each virtual host inside the `virtualhost` directive. For example by adding `ServerPath /mysite/` to www.mysite.com old browsers would be able to access the site by www.mysite.com/mysite/. Then you put the default page on the main server that politely tells them to get a new browser, and lists links to all the back doors of all the sites you host on that machine. When an old browser accesses the site they will be sent to the main page, and get a link to the correct page. New browsers will never see the main page and will go directly to the virtual hosts. You must remember to keep all of your links relative within the web sites, because the pages will be accessed from two different URL's (www.mysite.com and www.mysite.com/mysite/).

I hope I didn't lose you there, but its not an easy workaround. Maybe you should consider IP based hosting after all. A very similar workaround is also explained on the apache website at http://www.apache.org/manual/host.html.

If anyone has a great resource for Shared IP hosting, I would like to know about it. It would be nice to know what percent of browsers out there support HTTP 1.1, and to have a list of which browsers and versions support HTTP 1.1.

7.5 CGI scripts

There are two different ways to give your users CGI script capability. The first is make everything ending in `.cgi` a CGI script. The second is to make script directories (usually named `cgi-bin`). You could also use both methods. For either method to work the scripts must be world executable (`chmod 711`). By giving your users script access you are creating a big security risk. Be sure to do your homework to minimize the security risk.

I prefer the first method, especially for complex scripting. It allows you to put scripts in any directory. I like to put my scripts with the web pages they work with. For sites with allot of scripts it looks much better than having a directory full of scripts. This is simple to setup. First uncomment the `.cgi` handler at the end of the `srm.conf` file. Then make sure all your directories have the `option ExecCGI` or `All` in the `access.conf` file.

Making script directories is considered more secure. To make a script directory you use the ScriptAlias directive in the `srm.conf` file. The first argument is the Alias the second is the actual directory. For example `ScriptAlias /cgi-bin/ /usr/httpd/cgi-bin/` would make `/usr/httpd/cgi-bin` able to execute scripts. That directory would be used whenever someone asked for the directory `/cgi-bin/`. For security reasons you should also change the properties of the directory to `Options none, AllowOveride none` in the `access.conf` (just uncomment the example that is there). Also do not make your script directories subdirectories of your web page directories. For example if you are serving pages from `/home/httpd/html/`, don't make the script directory `/home/httpd/html/cgi-bin`; Instead make it `/home/httpd/cgi-bin`.

If you want your users to have there own script directories you can use multiple `ScriptAlias` commands. Virtual hosts should have there `ScriptAlias` command inside the `virtualhost` directive tags. Does anyone know a simple way to allow all users to have a cgi–bin directory without individual ScriptAlias commands?

7.6 Users Web Directories

There are two different ways to handle user web directories. The first is to have a subdirectory under the users home directory (usually `public_html`). The second is to have an entirely different directory tree for web directories. With both methods make sure set the access options for these directories in the `access.conf` file.

The first method is already setup in apache by default. Whenever a request for `/~bob/` comes in it looks for the `public_html` directory in bob's home directory. You can change the directory with the `UserDir` directive in the `srm.conf` file. This directory must be world readable and executable. This method creates a security risk because for Apache to access the directory the users home directory must be world executable.

The second method is easy to setup. You just need to change the `UserDir` directive in the `srm.conf` file. It has many different formats; you may want to consult the Apache documentation for clarification. If you want each user to have their own directory under `/home/httpd/`, you would use `UserDir /home/httpd`. Then when the request `/~bob/` comes in it would translate to `/home/httpd/bob/`. Or if you want to have a subdirectory under bob's directory you would use `UserDir /home/httpd/*/html`. This would translate to `/home/httpd/bob/html/` and would allow you to have a script directory too (for example `/home/httpd/bob/cgi-bin/`).

7.7 Daemon mode vs. Inetd mode

There are two ways that apache can be run. One is as a daemon that is always running (Apache calls this standalone). The second is from the inetd super–server.

Daemon mode is far superior to inetd mode. Apache is setup for daemon mode by default. The only reason to use the inetd mode is for very low use applications. Such as internal testing of scripts, small company Intranet, etc. Inetd mode will save memory because apache will be loaded as needed. Only the inetd daemon will remain in memory.

If you don't use apache that often you may just want to keep it in daemon mode and just start it when you need it. Then you can kill it when you are done (be sure to kill the parent and not one of the child processes).

To setup inetd mode you need to edit a few files. First in `/etc/services` see if http is already in there. If its not then add it:

```
http    80/tcp
```

Right after 79 (finger) would be a good place. Then you need to edit the `/etc/inetd.conf` file and add the line for Apache:

```
http    stream  tcp     nowait  root    /usr/sbin/httpd httpd
```

Be sure to change the path if you have Apache in a different location. And the second httpd is not a typo; the inet daemon requires that. If you are not currently using the inet daemon, you may want to comment out the rest of the lines in the file so you don't activate other services as well (FTP, finger, telnet, and many other things are usually run from this daemon).

If you are already running the inet deamon (`inetd`), then you only need to send it the SIGHUP signal (via kill; see kill's man page for more info) or reboot the computer for changes to take effect. If you are not running `inetd` then you can start it manually. You should also add it to your init files so it is loaded at boot (the `rc.local` file may be a good choice).

7.8 Allowing put and delete commands

The newer web publishing tools support this new method of uploading web pages by http (instead of FTP). Some of these products don't even support FTP anymore! Apache does support this, but it is lacking a script to handle the requests. This script could be a big security hole, be sure you know what you are doing before attempting to write or install one.

If anyone knows of a script that works let me know and I'll include the address to it here.

For more information goto Apacheweek's article at http://www.apacheweek.com/features/put.

7.9 User Authentication/Access Control

This is one of my favorite features. It allows you to password protect a directory or a file without using CGI scripts. It also allows you to deny or grant access based on the IP address or domain name of the client. That is a great feature for keeping jerks out of your message boards and guest books (you get the IP or domain name from the log files).

To allow user authentication the directory must have `AllowOverrides AuthConfig` set in the `access.conf` file. To allow access control (by domain or IP address) AllowOverrides Limit must be set for that directory.

Setting up the directory involves putting an `.htaccess` file in the directory. For user authentication it is usually used with an `.htpasswd` and optionally a `.htgroup` file. Those files can be shared among multiple `.htaccess` files if you wish.

For security reasons I recommend that everyone use these directives in there access.conf file:

```
<files ~ "/\.ht">
order deny,allow
deny from all
</files>
```

If you are not the administrator of the system you can also put it in your .htaccess file if AllowOverride Limit is set for your directory. This directive will prevent people from looking into your access control files (.htaccess, .htpasswd, etc).

There are many different options and file types that can be used with access control. Therefore it is beyond the scope of this document to describe the files. For information on how to setup User Authentication see the Apacheweek feature at http://www.apacheweek.com/features/userauth or the NCSA pages at http://hoohoo.ncsa.uiuc.edu/docs−1.5/tutorials/user.html.

7.10 su−exec

The su−exec feature runs CGI scripts as the user of the owner. Normally it is run as the user of the web server (usually nobody). This allows users to access there own files in CGI scripts without making them world writable (a security hole). But if you are not careful you can create a bigger security hole by using the su−exec code. The su−exec code does security checks before executing the scripts, but if you set it up wrong you will have a security hole.

The su−exec code is not for amateurs. Don't use it if you don't know what you are doing. You could end up with a gaping security hole where your users can gain root access to your system. Do not modify the code for any reason. Be sure to read all the documentation carefully. The su−exec code is hard to setup on purpose, to keep the amateurs out (everything must be done manually, no make file no install scripts).

The su−exec code resides in the `support` directory of the source. First you need to edit the `suexec.h` file for your system. Then you need to compile the su−exec code with this command:

```
gcc suexec.c -o suexec
```

Then copy the suexec executable to the proper directory. The Apache default is `/usr/local/etc/httpd/sbin/`. This can be changed by editing `httpd.h` in the Apache source and recompiling Apache. Apache will only look in this directory, it will not search the path. Next the file needs to be changed to user root (`chown root suexec`) and the suid bit needs to be set (`chmod 4711 suexec`). Finally restart Apache, it should display a message on the console that su−exec is being used.

CGI scripts should be set world executable like normal. They will automaticaly be run as the owner of the CGI script. If you set the SUID (set user id) bit on the CGI scripts they will not run. If the directory or file is world or group writable the script will not run. Scripts owned by system users will not be run (root, bin, etc.). For other security conditions that must be met see the su−exec documentation. If you are having problems see the su−exec log file named `cgi.log`.

Su−exec does not work if you are running Apache from inetd, it only works in daemon mode. It will be fixed in the next version because there will be no inetd mode. If you like playing around in source code, you can edit the http_main.c. You want to get rid of the line where Apache announces that it is using the su−exec wrapper (It wrongly prints this in front of the output of everything).

Be sure and read the Apache documentation on su−exec. It is included with the source and is available on the Apache web site at http://www.apache.org/docs/suexec.html

7.11 Imagemaps

Apache has the ability to handle server side imagemaps. Imagemaps are images on webpages that take users to different locations depending on where they click. To enable imagemaps first make sure the imagemap module is installed (its one of the default modules). Next you need to uncomment the `.map` handler at the end of the `srm.conf` file. Now all files ending in `.map` will be imagemap files. Imagemap files map different areas on the image to separate links. Apache uses map files in the standard NCSA format. Here is an example of using a map file in a web page:

```
<a href="/map/mapfile.map">
<img src="picture.gif" ISMAP>
</a>
```

In this example `mapfile.map` is the mapfile, and `picture.gif` is the image to click on.

There are many programs that can generate NCSA compatible map files or you can create them yourself. For a more detailed discussion of imagemaps and map files see the Apacheweek feature at http://www.apacheweek.com/features/imagemaps.

7.12 SSI/XSSI

Server Side Includes (SSI) adds dynamic content to otherwise static web pages. The includes are embedded in the web page as comments. The web server then parses these includes and passes the results to the web server. SSI can add headers and footers to documents, add date the document was last updated, execute a system command or a CGI script. With the new eXtended Server Side Includes (XSSI) you can do a whole lot more. XSSI adds variables and flow control statements (if, else, etc). Its almost like having an programming language to work with.

Parsing all HTML files for SSI commands would waste allot of system resources. Therefore you need to distinguish normal HTML files from those that contain SSI commands. This is usually done by changing the extension of the SSI enhanced HTML files. Usually the `.shtml` extension is used.

To enable SSI/XSSI first make sure that the includes module is installed. Then edit `srm.conf` and uncomment the `AddType` and `AddHandler` directives for `.shtml` files. Finally you must set `Options Includes` for all directories where you want to run SSI/XSSI files. This is done in the `access.conf` file. Now all files with the extension `.shtml` will be parsed for SSI/XSSI commands.

Another way of enabling includes is to use the `XBitHack` directive. If you turn this on it looks to see if the file is executable by user. If it is and `Options Includes` is on for that directory, then it is treated as an SSI file. This only works for files with the mime type text/html (`.html .htm` files). This is not the preferred method.

There is a security risk in allowing SSI to execute system commands and CGI scripts. Therefore it is possible to lock that feature out with the `Option IncludesNOEXEC` instead of Option Includes in the `access.conf` file. All the other SSI commands will still work.

For more information see the Apache mod_includes documentation that comes with the source. It is also available on the website at http://www.apache.org/docs/mod/mod_include.html.

For a more detailed discussion of SSI/XSSI implementation see the Apacheweek feature at http://www.apacheweek.com/features/ssi.

For more information on SSI commands see the NCSA documentation at http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.html.

For more information on XSSI commands goto ftp://pageplus.com/pub/hsf/xssi/xssi−1.1.html.

7.13 Module system

Apache can be extended to support almost anything with modules. There are allot of modules already in existence. Only the general interest modules are included with Apache. For links to existing modules goto the

Apache Module Registry at http://www.zyzzyva.com/module_registry/.

For module programming information goto http://www.zyzzyva.com/module_registry/reference/

8.Web Server Add−ons

Sorry this section has not been written yet.

Coming soon: mSQL, PHP/FI, cgiwrap, Fast−cgi, MS frontpage extentions, and more.

9.Intranet Section

In simple terms, the **Intranet** is the descriptive term being used for the implementation of Internet technologies within a corporate organisation, rather than for external connection to the global Internet. This implementation is performed in such a way as to transparently deliver the immense informational resources of an organisation to each individuals desktop with minimal cost, time and effort. This document attempts to explain in simple terms how to setup an Intranet using tools which are readily available and are generally costing little or are free.

This document assumes that you already know how to install TCP/IP on your Linux server and connect it physically to your LAN using an Ethernet network card. This also assumes you have some basic knowledge of Netware, WinNT and Mac systems. The configuration of the Netware server has been shown using version 3.1x as the basis. You can also use INETCFG to achieve the same result. On the client side the discussion is with respect to Windows 3.1x, Windows for Workgroups and Win95, WinNT and the Apple PowerMac.

I am using the private network addresses (RFC−1918) of 172.16.0.0 and 172.17.0.0 only as examples. You may choose suitable addresses depending on your configuration.

```
              Linux       Netware      WFWG/WinNT
              Server      Server       Server
             172.16.0.1   172.16.0.2   172.16.0.3
```

```
      |           |           |    172.16.0.0
------+-----+-----+--------+--+--------------
            |              | 172.16.0.254
         W/S 1          Router
       172.16.0.5          | 172.17.0.254
                           |
                 ----------+-------+--------
           172.17.0.0              |
                                 W/S 2
                               172.17.0.5
```

9.1 What is required

You will need the following software before attempting the installation.

- the HTTP server software which can be downloaded from OneStep NCSA HTTPd Downloader at http://hoohoo.ncsa.uiuc.edu/docs/setup/OneStep.html page.
- The Novell Netware Client available from HTTP://support.novell.com/ (The TCP/IP files are included with the client).
- The Microsoft TCP/IP client available from HTTP://www.microsoft.com/
- The Apple MacTCP client available from HTTP://www.apple.com/
- WWW Browsers like Netscape at HTTP://home.netscape.com/ or MS Internet Explorer at HTTP://www.microsoft.com/ or NCSA Mosaic from http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html

9.2 New versions of this document

New versions of the Linux Intranet Server HOWTO will be periodically posted to comp.os.linux.announce and comp.os.linux.help. They will also be uploaded to various Linux FTP sites, including sunsite.unc.edu.

The Latest version of this document is available in HTML format at http://www.linuxports.com/

9.3 Feedback

If you have questions or comments about this document, please feel free to mail Mr. Poet, at poet@linuxports.com. Suggestions, criticism and mail are always welcome. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanx.

10.Install the HTTP server

When you download the server you have two options: To get the source and compile it yourself, or get the precompiled binaries. The precompiled binaries for Linux (ELF) version are available at NCSA but not the older versions.

10.1 Preparation before downloading

The server at NCSA will guide you through the steps for configuration options and prepare the various files for you. But before you attempt to download HTTPd be prepared with answers to the following questions

The Operating System

First, you must choose whether to download the source or a pre–compiled version of the software. If your particular system doesn't appear in the menu, then you will have to get the default source, and compile it yourself.

To check the version of your Linux go to the command prompt on your Linux machine and type

```
linux:~$  uname -a
```

which will respond with a line which looks similar to this

```
linux:~$  uname -a
Linux linux 2.0.29 #4 Tue Sep 13 04:05:51 CDT 1994 i586
linux:~$
```

The version of Linux is 2.0.29.

The remaining parameters can be specified before downloading or configured later by modifying the file `srm.conf` in the `/usr/local/etc/httpd/conf` directory. The names of the actual directives that appear in the file `httpd.conf` are shown in brackets. The only exception is the directive DocumentRoot which appears in the file `srm.conf`

Process type (ServerType)

This specifies how your machine will run your HTTPd server. The preferred method is "standalone". This makes the HTTP daemon to be running constantly. If you choose to load HTTPd under "inetd", the server binary will be reloaded into memory for every request, which may slow your server down.

Binding Port (Port)

This specifies which port of your machine that the HTTPd daemon will bind to and listen for HTTP requests. If you can login as "root", use the default setting of 80. Otherwise choose a setting between 1025 and 65535.

Server user identity (User)

This is the user id the server will change to when answering requests and acting on files.This question needs to be answered only if you are running the server as "standalone". If you are someone without root permissions, just use your own login name. If you are system administrator, you might want to create a special user so you can control file permissions.

Server group identity (Group)

This is the group id the server will change to when answering requests and acting on files. This is similar to Server User identity and is applicable only if you are running the server as standalone.

If you do not have root permissions, just use the name of your primary group. You can find out your group by typing **groups** at the Linux command prompt.

Server administrator email address (ServerAdmin)

This is the email address that the user should send an email message to when reporting a problem with the server. You can put your personal e–mail address.

Location of server directory (ServerRoot)

This is where the server resides on your system. If you have root permissions leave it in its recommended location `/usr/local/etc/httpd`. If you cannot login as root, choose a subdirectory in your home path. You can find out the path of your home directory with the **pwd** command.

Location of HTML files (DocumentRoot)

This is where the HTML files to be served are located. The default location is `/usr/local/etc/httpd/htdocs`. You could however set it to be the home directory of the special user you chose in Server user identity, or a subdirectory in your home directory if you can't login as root.

When in doubt, use the default settings. Now that you have answers to the above questions you can Download NCSA HTTPd at http://hoohoo.ncsa.uiuc.edu/docs/setup/OneStep.html. You should read the HTTPd Documentation at http://hoohoo.ncsa.uiuc.edu/docs/ before you attempt installation. If you are planning to compile the code then you need to modify the makefiles in each of the th ree directories `support, src, cgi−src`. If your version of Linux is already supported then you just have to type **make linux** at the top level directory (i.e. `/usr/local/etc/httpd`)

10.2 Compiling HTTPd

Compiling is simple, just type `make linux` at the prompt in the server root directory. **Note:** Users of pre–ELF Linux have to uncomment `#define NO_PASS` in file `portability.h` and set `DBM_LIBS= -ldbm` in the `Makefile` before compiling HTTPd.

11.Testing HTTPd

After you have installed HTTPd, login as root and start it by typing **httpd &** . (assuming you have installed as standalone) You should now be able to see it in the list generated by **ps**. The simplest way to test HTTPd is by Telnet. At the Linux command prompt type

```
linux:~$  telnet 172.16.0.1 80
```

where 80 is the default port for HTTP. If you have configured "Port" as something different then type that number instead. You should get a response which looks like this

```
Trying 172.16.0.1...
Connected to linux.mydomain.
Escape character is '^]'.
```

Now if you type in any character and press Enter you should get a response similar to the one shown below.

```
HTTP/1.0 400 Bad Request
Date: Wed, 10 Jan 1996 10:24:37 GMT
Server: NCSA/1.5
Content-type: text/html

<HEAD><TITLE>400 Bad Request < /TITLE> < /HEAD>
<BODY><H1>400 Bad Request < /H1>
Your client sent a query that this server could
not understand.<P>
Reason: Invalid or unsupported method.<P>
< /BODY>
```

Now we are ready to connect to this server using another PC and a WWW Browser.

12.Connecting to the Linux Server

Please refer to the diagram shown in the chapter IntroIntroduction for the addressing scheme used. Workstation 1 (W/S1) is on network 172.16.0.0 and can access the Linux server directly whereas Workstation 2 (W/S2) is on network 172.17.0.0 and needs to use the gateway (router) 172.17.0.254 to access the Linux box. This gateway information needs to be provided while configuring the clients only on W/S2.

Netware refers to the gateway as 'ip_router'.

I am using W/S2 to illustrate the client setup. To setup W/S1 just change the address 172.17.0.5 to 172.16.0.5 and ignore all references to the gateway/router.

If you do not have a router you can skip the next section and proceed to

- Setup Netware Server if you use a Netware server.
- Setup MS Windows Client if you use the Microsoft Client.

12.1 Setup the Linux server

You may skip this section if you do not have a router.

You have to configure the Linux server to recognise the router thus allowing Workstation 2 to connect to the Web server. In order to setup the Linux server you should login as root. At the server prompt type

```
route add gw default 172.16.0.254
```

To use this gateway everytime you boot the Linux server edit the file `/etc/rc.d/rc.inet1` and change the line containing the gateway definition to `GATEWAY = "172.16.0.254"`. Make sure the line for adding the gateway is not commented out.

ALT: You can add routes to the networks on the other side of the router. This would be done as

```
route add -net 172.17.0.0 gw 172.16.0.254
```

To add this route everytime you boot Linux add the command to your `/etc/rc.d/rc.local` file.

12.2 Setup the Netware server

In order to setup the Netware server you should have Supervisor permissions or atleast Console operator permissions. If these cannot be got, try asking your Network Administrator to help you with the setup. At the Server enable the Ethernet_II frame type on the LAN by typing these commands or include them in the AUTOEXEC.ncf file.

```
load NE2000 frame=Ethernet_II name=IPNET
load TCPIP
bind IP to IPNET addr=172.16.0.2 mask=FF.FF.FF.0
```

You might have to specify the slot or board number while loading the NE2000 driver depending on your machine configuration. (eg: load NE2000 slot=3 frame=.....)

12.3 Setup the Netware Client

On the PC you have the choice of Win3.1,WFWG or Win95. The installation procedure differs between Win95 and the older windows if you are using the 32bit client from Microsoft or Novell. If you are going to use the 16bit client, the procedure is the same and you can refer to the Windows 3.x installation instructions. For installing the 32bit client for Win95 skip to Windows 95 installation.

Windows 3.x

If you are using Win3.1 or WFWG you can install the Netware Client (VLMs) and some additional files which are provided with the TCP/IP diskette, namely

TCPIP.exe, VTCPIP.386, WINSOCK.dll and WLIBSOCK.dll

Note that the WINSOCK.dll file is different from the ones provided with Win95 and Trumpet. Install the Netware Client with the support for windows. Copy VTCPIP.386, WINSOCK.dll and WLIBSOCK.dll to the SYSTEM directory and TCPIP.exe to the NWCLIENT directory. Now modify the STARTNET.bat in the NWCLIENT directory to

```
lsl
ne2000     ---> your network card driver
c:\windows\odihlp.exe     ---->if you are using WFWG
ipxodi
tcpip        ---> add this line
nwip         ---> if you use Netware/IP
vlm
```

Create a subdirectory (say) `\NET\TCP` and copy the files HOSTS, NETWORKS, PROTOCOLS and SERVICES from `/etc` on your Linux server or the directory `SYS:ETC` on your Netware server. Edit the copied HOSTS file to add the line for your new Linux server. This will enable you to refer to the Linux server as `http://linux.mydomain/` instead of `http://172.16.0.1/`in your WWW browser

```
127.0.0.1      localhost
172.16.0.1      linux.mydomain
```

Edit the NET.cfg file in NWCLIENT directory

```
   Link Driver NE2000
       port 300
       int 3
       MEM D0000
       FRAME Ethernet_802.2

; ---- add these lines ----

      FRAME Ethernet_II

   Protocol TCPIP
```

```
    PATH TCP_CFG C:\NET\TCP
    ip_address  172.17.0.5
    ip_netmask  255.255.255.0
    ip_router   172.17.0.254     ---> add the address of your gateway only
                                ---> if you have to use this
                                ---> gateway to reach your HTTP server

 Link Support
    MemPool 6192       ---> the minimum is 1024. Try with different values
    Buffers 10 1580    ---> this again can be fine tuned

;---------------------------------
; You may need to add lines like these if you are using Netware/IP
;
 NWIP
    NWIP_DOMAIN_NAME  mydomain
    NSQ_BROADCAST   ON
    NWIP1_1 COMPATIBILITY   OFF
    AUTORETRIES     1
    AUTORETRY SECS  10
```

Edit the SYSTEM.ini file in the WINDOWS directory and add this entry for VTCPIP.386

```
[386Enh]
.....
network=*vnetbios, vipx.386, vnetware.386, VTCPIP.386
.....
```

Reboot your PC, run STARTNET.bat and you can now use your favorite WWW browser to access your Web pages. You need not login to Netware and you don't have to run TCPMAN (if you use Trumpet Winsock).

Windows 95

This section explains how to install the 32bit client on Win95. Firstly you must install the following

```
Client for Netware Networks (from Microsoft or Novell)
Microsoft TCP/IP Protocol
Network Adapter
```

To install these items, click on My Computer, Control Panel, Networks. Click Add. You will now be in a window that displays Client, Adapter, Protocol and Service. To install the Client for Netware Networks:

```
1. DoubleClick on Client
2. Click on Microsoft or Novell
3. DoubleClick on Client for Netware Networks
```

To install the TCP/IP Protocol:

```
1. DoubleClick on Protocol
2. Click on Microsoft
3. DoubleClick on TCP/IP
```

Windows 95 by default installs several other protocols automatically. Remove them by clicking on them and clicking the Remove button. Typically Win95 installs the Microsoft NetBeui protocol, and IPX/SPX compatible protocol. You can delete the NetBEUI protocol, but you will need the IPX/SPX protocol if you wish to login to the Netware Server.

To setup TCP/IP click on TCP/IP, click on Properties, click on the tab IP address

```
Enter your IP address in the "Specify an IP address "
                      box as 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
```

select the tab Gateway

```
Enter your gateway (router) address in the box New gateway
   as 172.17.0.254
Click the Add button
```

The gateway address should now appear under the installed gateways box. Now Click OK.

You should get a message to reboot. Do so. You should now be able to use the Browser to connect to your HTTP Server.

12.4 Setup Microsoft Client

If you are using the Microsoft Client for accessing your network, then this section details how to install TCP/IP for

- Windows for Workgroups (WFWG)
- Windows 95
- Windows NT 4.0

Note: To enable you to refer to the Linux server as `http://linux.mydomain/` instead of `http://172.16.0.1/` in the WWW browser and all your intranet commands you need to edit the `hosts` file. You can add more entries for each of your other hosts (Netware, Unix, WinNT) as well. The Windows family keeps its HOSTS file in `\WINDOWS` or in `\WINDOWS\SYSTEM` depending on the version. Edit this file and add a line for your Linux server as:

```
127.0.0.1      localhost
172.16.0.1     linux.mydomain

172.16.0.2     netware.mydomain
172.16.0.3     winNT.mydomain
172.16.0.5     ws_1
```

Windows for Workgroups

This section explains how to install the 32bit client on WFWG. Firstly you must download the TCP/IP drivers for Windows from Microsoft. The current version is 3.11b and is available at ftp://ftp.microsoft.com or other sites as `tcp32b.exe`. Make sure that you have load Win32s before trying to load the TCP/IP–32bit driver.

Having expanded the TCP/IP files into a temporary directory (say `C:\TEMP`), check your `\WINDOWS\SYSTEM` directory for copies of `OEMSETUP.INF`. If there are any, rename them. Now copy the `OEMSETUP.INF` file from the TEMP directory to the `\WINDOWS\SYSTEM` directory. If you have loaded any other TCP/IP stacks on your system, please remove them before you proceed.

Start Network Setup or Windows Setup/Change Network settings

```
Click the Networks button
Click Install Microsoft Windows Network.
  Choose support for additional networks (if required)
Click OK
```

You should be prompted for your network adapter – select the appropriate one. If you are not prompted, then

```
Click the Adapter button
    select an adapter (say NE2000)
    Click OK
Click the Protocol button
    select the MS TCP/IP-32 protocol
    click OK
```

You will now be prompted to configure the TCP/IP protocol stack. You can always reconfigure this by highlighting the TCP/IP protocol shown in the box Adapters and clicking the Setup button.

```
In the IP address box enter 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
Enter your gateway (router) address in the box default gateway
  as 172.17.0.254
```

Click OK. The computer will ask you to restart. Do so. You should now be able to use the Browser to connect to your HTTP Server.

Windows 95

This section explains how to install the 32bit client for Microsoft on Win95. Firstly you must install the following

```
Client for Microsoft Networks
Microsoft TCP/IP Protocol
Network Adapter
```

To install these items, click on My Computer, Control Panel, Networks. Click Add. You will now be in a window that displays Client, Adapter, Protocol and Service. To install the Client for Microsoft Networks:

```
1. DoubleClick on Client
2. Click on Microsoft
3. DoubleClick on Client for Microsoft Networks
```

To install the TCP/IP Protocol:

```
1. DoubleClick on Protocol
2. Click on Microsoft
3. DoubleClick on TCP/IP
```

Windows 95 by default installs several protocols automatically. Remove them by clicking on them and clicking the Remove button. Typically Win95 installs the Microsoft NetBeui protocol.

To setup TCP/IP click on TCP/IP, click on Properties, click on the tab IP address

```
Enter your IP address in the "Specify an IP address "
                      box as 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
```

select the tab Gateway

```
Enter your gateway (router) address in the box New gateway
   as 172.17.0.254
Click the Add button
```

The gateway address should now appear under the installed gateways box. Now Click OK.

You should get a message to reboot. Do so. You should now be able to use the Browser to connect to your HTTP Server.

Windows NT

This section details how to Install the TCP/IP client for WinNT 4.0. Start Control Panel/ Network

```
Select the Adapter tab.
    Click Add to add a new adapter (if you don't have one)
```

You should be prompted for your network adapter – select the appropriate one. To add the protocols.

```
Select the Protocols tab
    Click Add
    Select the TCP/IP protocol
    Click OK
```

You will now be prompted to configure the TCP/IP protocol stack. You can always reconfigure this by highlighting the TCP/IP protocol and clicking the Properties button.

```
Select the tab IP Address
   Mark the checkbox 'Specify an IP address'
   In the IP address box enter 172.17.0.5
   In the Subnet Mask box enter 255.255.255.0
   Enter your gateway (router) address in the box Default Gateway
       as 172.17.0.254
```

Click OK. The computer will ask you to restart. You can now use any Browser to connect to your HTTP Server.

12.5 Setup TCP/IP on Macintosh

If you are using the Macintosh for accessing your network, then this section details how to install MacTCP for the PowerMacs.

Note: To enable you to refer to the Linux server as `http://linux.mydomain/` instead of `http://172.16.0.1/` in the WWW browser and all your intranet commands you need to edit the `hosts` file. The format of the hosts file is different from the one used in Unix. The Mac hosts file is based on RFC−1035. You can add more entries for each of your other hosts (Netware, Unix, WinNT) as well. The MacOS keeps its HOSTS file in the `Preferences folder` under the `System folder`. Edit this file and add a line for your Linux server as:

```
linux.mydomain    A  172.16.0.1

netware.mydomain  A  172.16.0.2
winNT.mydomain    A  172.16.0.3
ws_1              A  172.16.0.5
```

MacTCP

This section explains how to install MacTCP. Firstly you must download the MacTCP files from Apple or install it from the Internet Connection CD. To configure MacTCP, click the Apple Menu/ Control Panels/ TCP/IP. In the screen change the setting for 'Connect via:' to 'Ethernet'

Change the 'Configure' setting to 'Manually'

```
In the IP address box enter 172.17.0.5
In the Subnet Mask box enter 255.255.255.0
Enter your gateway (router) address in the box
  Router address as 172.17.0.254
```

Click OK. You should now be able to use the Browser to connect to your HTTP Server.

13.Setting up the Intranet

An Intranet cannot be complete without sharing the resources on the different platforms. You will need support for other filesystems, so that you can access the data available on them. This document provides instructions to connect Linux to the following popular filesystems.

- NCP filesystem for Netware
- SMB filesystem for Windows
- NFS filesystem for Unix

These filesystems can be compiled into the Linux kernel or added as modules, depending on the version of Linux. If you are not familiar with compiling the kernel you can refer to the Kernel HOWTO http://sunsite.unc.edu/mdw/HOWTO/Kernel−HOWTO.html and the Module HOWTO http://sunsite.unc.edu/mdw/HOWTO/Module−HOWTO.html for compiling the kernel with modules.

13.1 NCPFS

To share the files on the Netware server you will need support for NCP (ncpfs). NCPFS works with kernel version 1.2.x and 1.3.71 upwards. It does not work with any earlier 1.3.x kernel. It cannot access the NDS database in Netware 4.x, but can make use of the bindery. If you are using Netware 4.x you can enable bindery support for specific containers using the command `Set Bindery Context` at the console as:

```
set Bindery Context = CORP.MYDOM;WEBUSER.MYDOM
```

In the above example two containers have bindery support enabled.

You will need to download the NCP filesystem utilities using the URL ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/ncpfs.tgz (currently ncpfs−2.0.10) from Sunsite.

Installation

To install the ncpfs utilities, type

```
zcat ncpfs.tgz | tar xvf -
```

to expand the files into its own directory. In this case you will get a directory `ncpfs-2.0.10` Change your directory to this ncpfs directory before proceeding with the installation. Read the README and edit the Makefile if necessary.

The installation of ncpfs depends on the kernel version you are using. For kernel 1.2, you should simply type 'make'. Subsequently typing 'make install' will install the executables and man pages.

If you use Kernel 1.3.71 or later, you might have to recompile your kernel. With these kernels, the kernel part of ncpfs is already included in the main source tree. To check if the kernel needs to be recompiled type

```
cat /proc/filesystems
```

It should show you a line saying that the kernel knows ncpfs.

If ncpfs is not there, you can either recompile the kernel or add ncpfs as a module. For recompiling the kernel you should type 'make config' and when it asks you for

```
The IPX protocol (CONFIG_IPX) [N/y/?]
```

simply answer 'y'. Probably you do not need the full internal net that you are asked for next. Once the kernel is successfully installed, reboot, check `/proc/filesystems` and if everything is OK proceed with the installation of the ncpfs utilities. Change directory to the location holding your downloaded ncpfs files, and type 'make'. After the compilation is finished type 'make install' to install the various utilities and man pages.

Mounting NCPFS

To check the installation type

```
   ipx_configure --auto_interface=on --auto_primary=on

....wait for 10 seconds and type

   slist
```

You should be able to see a list of your Netware servers. Now we are ready to share files from the Netware server.

Suppose we need to access HTML files from directory `\home\htmldocs` on volume VOL1: on the server MYDOM_NW, I recommend that you create a new user (say) 'EXPORT' with password 'EXP123' on this server to whom you grant appropriate access rights to this directory using SYSCON or NWADMIN.

On the Linux machine create a new directory `/mnt/MYDOM_NW`. Now type the command

```
ncpmount -S MYDOM_NW -U EXPORT -P EXP123 /mnt/MYDOM_NW
```

to mount the netware file system. Typing the command

```
ls /mnt/MYDOM_NW/vol1/home/htmldocs
```

will show you a list of all the files in `MYDOM_NW/VOL1:\HOME\HTMLDOCS` (using Netware file notation). If you have any problems please read the IPX HOWTO at http://sunsite.unc.edu/mdw/HOWTO/IPX–HOWTO.html for more insights into the IPX system.

13.2 SMBFS

To share the files on the Windows server you will need support for SMB (smbfs).

You will need to download the SMB filesystem utilities from ftp://sunsite.unc.edu/pub/Linux/system/filesystems/smbfs/smbfs.tgz (currently smbfs−2.0.1) from Sunsite.

Installation

To install the smbfs utilities, type

```
zcat smbfs.tgz | tar xvf -
```

to expand the files into its own directory. In this case you will get a directory `smbfs−2.0.1` Change your directory to this smbfs directory before proceeding with the installation. Read the README and edit the Makefile if necessary.

The installation of smbfs depends on the kernel version you are using. For kernel 1.2, you should simply type 'make'. Subsequently typing 'make install' will install the executables and man pages.

If you use Kernel 2.0 or later, you might have to recompile your kernel. With these kernels, the kernel part of smbfs is already included in the main source tree. To check if the kernel needs to be recompiled type

```
cat /proc/filesystems
```

It should show you a line saying that the kernel knows smbfs.

If smbfs is not there, you can either recompile the kernel or add smbfs as a module. For recompiling the kernel you should type 'make config' and when it asks you for adding SMB filesystem support simply answer yes. Once the kernel is successfully installed, reboot, check `/proc/filesystems` and if everything is OK proceed with the installation of the smbfs utilities. Change directory to the location holding your downloaded smbfs files, and type 'make'. After the compilation is finished type 'make install' to install the various utilities and man pages.

Mounting SMBFS

In our example let us assume that the WinNT server is called 'MYDOM_NT' and is sharing its directory `C:\PUB\HTMLDOCS` with a share name of 'HTMLDOCS' without a password. On the Linux machine create a new directory `/mnt/MYDOM_NT`. Now type the command

```
smbmount //MYDOM_NT/HTMLDOCS /mnt/MYDOM_NT -n
```

to mount the SMB (windows share) file system. If this does not work try

```
smbmount //MYDOM_NT/COMMON /mnt/MYDOM_NT -n -I 172.16.0.3
```

Typing the command

```
ls /mnt/MYDOM_NT
```

will show you a list of all the files in `bsol;bsol;MYDOM_NT\PUB\HTMLDOCS` (using Windows file notation).

13.3 NFS

First you will need a kernel with the NFS file system either compiled in or available as a module.

Suppose you have a Unix host running NFS with the name MYDOM_UNIX and an IP address of 172.16.0.4. You can check the directories that are being exported (shared) by this host by typing the command

```
showmount -e 172.16.0.4
```

Once we know the exported directories you can mount them by entering a appropriate mount command. I recommend that you create a subdirectory under '/mnt' (say) 'MYDOM_UNIX' and use that as your mount point.

```
mount -o rsize=1024,wsize=1024 172.16.0.4:/pub/htmldocs /mnt/MYDOM_UNIX
```

The rsize and wsize may have to be changed depending on your environment.

If you have any problems please read the NFS HOWTO at http://sunsite.unc.edu/mdw/HOWTO/NFS−HOWTO.html for more insights into the NFS system.

14.Accessing the Web

Now that we have setup the HTTP server, the clients and interconnected the Linux server with the other servers, we need to make some small adjustments on the Linux server to be able to access these mounted filesystems from the Web Browser.

14.1 Accessing the mounted filesystems

To access the mounted directories in your HTML pages you have two methods:

- Create a link in DocumentRoot (`/usr/local/etc/httpd/htdocs`) to refer to the mounted directory as

```
ln -s /mnt/MYDOM_NW/vol1/home/htmldocs netware
                  or

ln -s /mnt/MYDOM_NT    winNT
                  or

ln -s /mnt/MYDOM_UNIX    unix
```

- to edit the file `srm.conf` in your `/usr/local/etc/httpd/conf` directory and add a new alias.

```
# Alias fakename realname
Alias /icons/     /usr/local/etc/httpd/icons/

# alias for netware server
Alias /netware/   /mnt/MYDOM_NW/vol1/home/htmldocs/
Alias /winNT/     /mnt/MYDOM_NT/
Alias /unix/      /mnt/MYDOM_UNIX
```

And restart your HTTPd. You can access the documents on the netware server by referring to them as `http://linux.mydomain/netware/index.htm` for the netware files and similar notations for the others.

14.2 Connecting to the Internet

You can finally connect your Intranet to the Internet to access E–Mail and all the wonderful information out there. I propose to write a brief note on how to do this in a future revision. Detailed explanations are available in the ISP Hookup HOWTO from http://sunsite.unc.edu/mdw/HOWTO/ISP–Hookup–HOWTO.html and Diald mini HOWTO at http://sunsite.unc.edu/mdw/HOWTO/mini/Diald for setting up these connections.

14.3 Other uses

The HTTP server can be used in the office to provide transparent access to information residing on different servers, at several locations and directories. The data can be simple documents in Word, Lotus spreadsheets, or complex databases.

The application of this technology is being typically used as follows:–

- Publishing corporate documents

  These documents can include newsletters, annual reports, maps, company facilities, price lists, product information literature, and any document which is of value within the corporate entity.
- Access into searchable directories

  Rapid access to corporate phone books and the like. This data can be mirrored at a Web site or, via CGI scripts, the Web server can serve as a gateway to back−end pre−existing or new applications. This means that, using the same standard access mechanisms, information can be made more widely available and in a simpler manner. This means that it can be used to create an interface with RDBMS like ORACLE and SYBASE for generating real−time information. Here is a list of links to such sites on the Web.

  − Web Access − http://cscsun1.larc.nasa.gov/~beowulf/db/web_access.html − CGI gateways − HTTP://www.w3.org/hypertext/WWW/RDBGate/Overview.html

- Corporate/Department/Individual pages

  As cultures change within organistions to the point where even each department moves towards their own individual mission statements, the Intranet technology provides the ideal medium to communicate current information to the Department or Individual. Powerful search engines provide the means for people to find the group or individual who has the answers to the continuous questions which arise in the normal day−to−day course of doing business.
- Simple Groupware applications

  With HTML forms support, sites can provide sign−up sheets, surveys and simple scheduling.
- Software distribution

  Administrators can use the Intranet to deliver software and up−dates on−demand to users across the corporate network . This can be done with 'Java' which allows the creation and transparent distribution of objects on−demand rather than just data or applications. This is indeed possible more easily with the newer versions of Linux which has builtin support for Java.
- Mail

  With the move to the use of Intranet mail products with standard and simple methods for attachment of documents, sound, vision and other multimedia between individuals, mail is being pushed further forward as a simple, de facto communications method. Mail is essentially individual to individual, or individual to small group, communication. Several utilities are available on the Linux platform to setup an E−mail system like **sendmail, pop3d, imapd**.

- User Interface

  The Intranet technology is evolving so rapidly that the tools available, in particular HTML, can be used to dramatically change the way we interface with systems. With HTML you can build an Interface which is only limited by the creators imagination. The beauty about using Intranet technologies for this is that it is so simple. Clicking a hyperlink from HTML can take you to another page, it could ring an alarm, run a yearend procedure or anything else that a computer program can do.

15.More things to do

Here is a list of other interesting things to do with your Linux Intranet server. All the software mentioned below is freeware or shareware.

- Browse the Linux server using Network Neighbourhood in Win95/ NT; Setup a WINS like NBT server. Check out the SAMBA Web page at http://lake.canberra.edu.au/pub/samba/samba.html
- Implement a search engine on your Intranet. Connect to ht://Dig at http://htdig.sdsu.edu/
- Use CUSeeMe by setting up a local reflector. Refer to their page at Cornell http://cu−seeme.cornell.edu/
- Setup Web Conferencing. Use COW from http://thecity.sfsu.edu/COW/
- Deploy a SQL database. Refer to the mSQL Home page at http://Hughes.com.au/
- Setup FTP,Gopher,Finger,Bootp servers on the Netware server. Get them at http://mft.ucs.ed.ac.uk/
- Emulate a Netware server. Check out the NCP Utilities at ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/

If you find other interesting things to do with your Linux Intranet server, please feel free to mail me.

16.Credits and Legalities

16.1 Thanks

Thanks to the people at NCSA for providing such excellent documentation, David Anderson and all others for trying out this HOWTO and sending in their comments. The details on Netware/IP are courtesy Romel Flores (rom@mnl.sequel.net).

16.2 Copyright information

linux−howto@sunsite.unc.edu Finger for phone number and snail mail address.

17.For further reading

17.1 O'Reilly & Associates Books

In my humble opinion O'Reilly & Associates make the best technical books on the planet. They focus mainly on Internet, Unix and programming related topics. They start off slow with plenty of examples and when you finish the book your an expert. I think you could get by if you only read half of the book. They also add some humor to otherwise boring subjects.

They have great books on HTML, PERL, CGI Programming, Java, JavaScript, C/C++, Sendmail, Linux and much much more. And the fast moving topics (like HTML) are updated and revised about every 6 months or so. So visit the O'Reilly & Associates web site or stop by your local book store for more info.

And remember if it doesn't say O'Reilly & Associates on the cover, someone else probably wrote it.

17.2 Internet Request For Comments (RFC)

- RFC1866 written by T. Berners−Lee and D. Connolly, "Hypertext Markup Language – 2.0", 11/03/1995
- RFC1867 writtenm by E. Nebel and L. Masinter, "Form−based File Upload in HTML", 11/07/1995
- RFC1942 written by D. Raggett, "HTML Tables", 05/15/1996
- RFC1945 by T. Berners−Lee, R. Fielding, H. Nielsen, "Hypertext Transfer Protocol −− HTTP/1.0", 05/17/1996.
- RFC1630 by T. Berners−Lee, "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World−Wide Web", 06/09/1994
- RFC1959 by T. Howes, M. Smith, "An LDAP URL Format", 06/19/1996