**Linux System Basics**

**Linux is a multitasking, multiuser operating system, which means that many people can run many different applications on one computer at the same time. This differs from MS-DOS, where only one person can use the system at any one time.**

Linux probably has the broadest reach of any operating system, from tiny systems the size of phone jacks, to cell phones, to supercomputer clusters bigger than your high school. It has infiltrated the fields of telecommunications, embedded systems, satellites, medical equipment, military systems, computer graphics, and—last but not least—desktop computing.

The NASA Center for Computational Sciences (NCCS) at the Goddard Space Flight Center does. Its Linux-based high-performance computing (HPC) clusters are designed to dramatically increase throughput for applications ranging from studying weather and climate variability to simulating astrophysical phenomena. Linux supplements NCCS architecture designed to scale to as many as 40 trillion floating-point operations per second (TFLOPS) in its full configuration.

Linux runs more of the world’s top supercomputers than any other operating system. In fact, as of this writing Linux runs an astonishing 75 percent of the top 500 supercomputers on the planet.\* According to department heads at the Lawrence Livermore National Laboratory in Livermore, CA, Linux runs 10 of their massive systems, all of which are on the TOP500 List. Those systems include BlueGene/L, the world’s most powerful supercomputer, and Thunder, which currently ranks nineteenth

**Administrator’s jobs’ responsibilities:**

• Administer and manage large Linux server environment, with an emphasis on performance monitoring, tuning, and management.

• Oversee database physical design, administration, and documentation.

• Provide network troubleshooting, escalated service desk support, and proactive monitoring of mission-critical systems

• Provide guidance and direction of technology solutions for the organization; train and mentor junior-level administrators.

• Supply daily technical support and on-call consulting advice for the hardware and operating system environment supporting the collection platform; administer Linux server infrastructure to maintain stability as well as maximize efficiencies in the computing environment.

• Install, configure, and troubleshoot all hardware, peripherals, and equipment necessary to meet integrated systems objectives; provide support functions on escalated issues.

• Provide effective first/second-level support for a company’s Linux environment across 300-plus servers, including Linux blades.

• Manage all aspects of the integrity of the environment, including security, monitoring (capacity and performance), change control, and software management.

• Interface with other internal support groups such as Change Control, Application Development, Engineering, Database Administrators, Web Services, Storage, Security, Operations, and Command Centers.

• Administer infrastructure services—DNS, NIS, LDAP, FTP, SMTP, Postfix/ Sendmail, NFS, Samba—and application and database servers, with an emphasis on automation and monitoring.

**SETTING UP A LINUX MULTIFUNCTION SERVER**

system will deploy:

• A web server (Apache 2.0.x)

• A mail server (Postfix)

• A DNS server (BIND 9)

• An FTP server (ProFTPD)

• Mail delivery agents (POP3/POP3s/IMAP/IMAPs)

• Webalizer for web site statistics

**Server Requirements**

* use almost any distribution of Linux to configure a web server. We chose Debian because we wanted to use a stable distribution of Linux
* To set up a Linux Internet server, you will need a connection to the Internet and a static IP address. If you cannot obtain a static IP address, you can set up the system with the address leased to you by your ISP and configure it statically. Make sure you know how long the lease runs, in case you have to change the IP address while your system is running.
* need a computer with at least a Pentium III CPU, a minimum of 256 MB of RAM, and a 10 GB hard drive.

**Installing Debian**

* After you boot into the Debian CD-ROM disk, you will see a login screen. Make sure to type in linux26 to get the most recent Version 2.6 kernel instead of the older version 2.4
* The installer will guide you through a series of installation screens. When you reach the screen called “Configure the Network,” Debian first suggests **configuring your network with DHCP**.
* You can do that if you have DHCP available. If you do not, Debian will default to a screen that allows you to configure your network manually.
* You will be asked to provide the hostname of the server, a domain name, a gateway, an IP address, a netmask, and a nameserver. If you have a registered domain and a static IP address, you’re ready to go. If you don’t have a registered domain name, you will need one.
* You will reach the hard disk partitioning screens. For the purposes of this book, just create one big partition with the mount point / (just a slash) and a swap partition. Choose the option to put all files in one partition. Finally, choose the finish partitioning option and write the results to disk.
* Proceed through the few remaining installation screens, which eventually ask you to reboot to initialize the kernel and finish the installation.
* After the reboot, Debian will want you to add a nonprivileged user during installation. That allows you to log in and use the su command to become root. For security reasons, system administrators have established a standard practice of not logging into the system as root unless they need to recover a failed system
* Name your first user account Administrator and give it a user ID of admin
* When you reach the Debian software selection screen, move your cursor to the box next to “mail server,” press the Space bar, and let the system install the default packages until you reach an option where you see the libc client.
* You should install the libc client with regular Unix mailbox support rather than maildir support
* Debian will also want you to configure Exim as the mail transfer agent (MTA)
* on the last screen involved with configuring Exim, enter the username admin as the email recipient for root and postmaster

**Logging in Remotely**

should log into the server from a remote console on your desktop.

u do further administration from another system (even a laptop), because a secure server normally runs in what is called headless mode—that is, it has no monitor or keyboard

On the remote machine you need only an SSH client,

**Configuring the Network**

If you used DHCP during the Debian installation, you should now configure your server with a static IP address

To change the settings to use a static IP address, you’ll need to become root and edit the file /etc/network/interfaces to suit your needs. As an example, we’ll use the IP address 70.153.258.42.

**Our configuration file starts out looking like this:**

# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface

auto lo

iface lo inet loopback

# The first network card - this entry was created during the Debian

# installation

# (network, broadcast, and gateway are optional)

# The primary network interface

iface eth0 inet dhcp

After editing the /etc/network/interfaces file, restart the network by entering: # /etc/init.d/networking restart You will then need to edit /etc/resolv.conf and add nameservers to resolve Internet hostnames to their corresponding IP addresses. Our resolv.conf looks as follows:

search server

nameserver 70.153.258.42

nameserver 70.253.158.45

nameserver 151.164.1.8

**Changing the Default Debian Packages**

Use the following commands to modify the startup scripts:

# update-rc.d -f exim remove Removing any system startup links for /etc/init.d/exim ...

**Setting Up Quotas**

install the quota packages using apt-get: # apt-get install quota quotatool

just add the usrquota and grpquota options to the partition with the mount point /:

run the following commands to add files to the root directory:

# touch /quota.user /quota.group

# chmod 600 /quota.\*

# mount -o remount /

# quotacheck –avugm

**Providing Domain Name Services**

* Set up a minimal configuration for BIND, the ubiquitous DNS server.
* Debian provides a stable version of BIND in its repositories. We’ll install and set up BIND and secure it in a chroot environment, meaning it won’t be able to see or access files outside its own directory tree. This is an important security technique. The term chroot refers to the trick of changing the root filesystem (the / directory) that a process sees, so that most of the system is effectively inaccessible to it.
* To install BIND on your Debian server, run this command: # apt-get install bind9

Setting up bind9 (9.2.4-1)

Adding group `bind' (104) Done.

Adding system user `bind'

Adding new user `bind' (104) with group `bind'

stop the service by running the following command: # /etc/init.d/bind9 stop

**Adding a Relational Database: MySQL**

Web sites and web service applications use relational databases to embed objects into web pages.

Web browsers can stimulate 30 requests at once, increasing loads on CPUs, memory, and disk access.

To make effective use of the database, you will need to know how to:

1. Install and start MySQL.

2. Create a MySQL root user.

3. Create a regular MySQL user, which the application will use to access the database.

4. Perform backups and restorations of databases.

To install the database server, a convenient client program that you can use to administer the server, and the library needed by both, issue this command:

# apt-get install mysql-server mysql-client libmysqlclient12-de

Create the MySQL root user by entering:

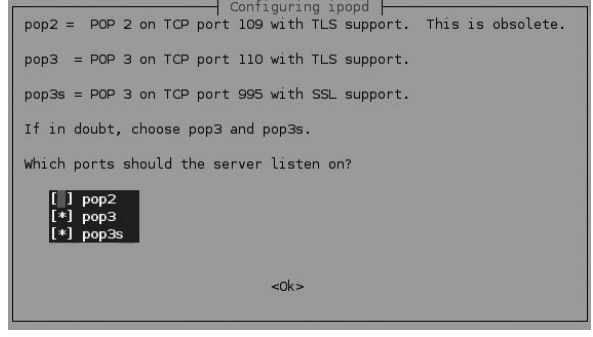
# mysqladmin -u root password 'pword'

**Configuring Mail Securely with Postfix, POP3, and IMAP**

* Sendmail has served as the Internet’s primary MTA. Many applications written for Linux expect to find Sendmail running on the server. Written before the Internet became open to the public
* MTAs such as Postfix and Exim must be able to appear to applications as if they are Sendmail
* use Postfix rather than Sendmail as our MTA.
* Securing email involves keeping unauthorized users off the server altogether (so they can’t use it to send unsolicited bulk email), making sure that nobody can spoof legitimate users, and protecting the content of each email from being snooped on or changed in transit.
* To promote authentication, we will install Postfix with Transport Layer Security (TLS), a protocol better known as the Secure Sockets Layer (SSL).
* To install the packages needed by Postfix and the other mail components, enter:

# apt-get install postfix postfix-tls libsasl2 sasl2-bin \

* libsasl2-modules ipopd-ssl uw-imapd-ssl



**DOMAIN NAME SYSTEM**

***D***omain***N***ame***S***ystem (or ***S***ervice or ***S***erver), an [Internet](http://www.webopedia.com/TERM/I/Internet.html) services - [hierarchical](http://en.wikipedia.org/wiki/Hierarchical) distributed naming system for computers, services, or any resource connected to the [Internet](http://en.wikipedia.org/wiki/Internet) or a [private network](http://en.wikipedia.org/wiki/Private_network) that translates [domain names](http://www.webopedia.com/TERM/D/domain_name.html) into IP addresses.

The Domain Name System distributes the responsibility of assigning domain names and mapping those names to IP addresses by designating authoritative name servers for each domain.

Authoritative name servers are assigned to be responsible for their supported domains, and may delegate authority over sub-domains to other name servers.

This mechanism provides distributed and fault tolerant service and was designed to avoid the need for a single central database.

DNS is the world’s largest distributed digital directory. Like an online telephone directory, you use it to match names with numbers—but with DNS, the numbers are the IP addresses of the multitude of servers connected to the Internet, including those that manage small web sites and gigantic server farms like Google and Amazon.

**Advantages of Localized DNS Administration**

* It gives you total control over which systems host your public services (e.g., web services and email), and putting DNS into your infrastructure allows you more scalability:
* You can add servers as needed and even do load balancing among them.
* This becomes important if you own and operate several active domains or internal authentication services.
* You also have more control over keeping your names updated.

**Getting into the BIND**

Most of the DNS servers in the world are run by the Berkeley Internet Name Daemon, or BIND. BIND is standard on every version of Unix and Linux

**Components of BIND**

BIND comes with three components.

The first is the service or daemon that runs the answering side of DNS. This component is called **named** (pronounced name-dee). It answers the phone when it rings.

The second item in the BIND bundle is the **resolver library**

BIND uses its own little directory called resolv.conf

the resolv.conf file looks like on computers in the centralsoft.org domain:

search centralsoft.org

nameserver 70.253.158.42

nameserver 70.253.158.45

The third part of BIND provides tools such as the dig command for testing DNS

**Setting Up a DNS Server**

After the initial stages of the Debian install, you’ll see a graphic screen asking you to choose the type of installation you want. The screen will look like this:

( ) Desktop Environment

( ) Web Server

( ) Print Server

( ) DNS Server

( ) File Server

( ) Mail Server

( ) SQL database

( ) manual package selection

To install BIND on your Debian server, run the command:

# apt-get install bind9

**Using a chroot Environment for Security**

Many security administrators recommend running BIND as a non-root user in an isolated directory called a chroot environment

To put BIND in a chroot environment, you need to create a directory where the service can run unexposed to other processes.

This directory will contain all the files that BIND needs, and it will look like the whole filesystem to BIND after you issue the chroot command.

To provide a complete environment for running BIND, create the necessary directories under /var/lib:

# mkdir -p /var/lib/named/etc

# mkdir /var/lib/named/dev

# mkdir -p /var/lib/named/var/cache/bind

# mkdir -p /var/lib/named/var/run/bind/run

Then move the config directory from /etc to /var/lib/named/etc:

# mv /etc/bind /var/lib/named/etc

**Configuring an Authoritative DNS Server**

* A system administrator has to come forward with the domain name and number (IP address) and make them part of the distributed DNS directory. Administrators do this by creating listings in what DNS aficionados call zone files.
* A zone holds the information for a domain
* BIND’s versatility allows you to manage several DNS servers at once, and to manage multiple domains independently on one server.
* Each web site is in a different domain, so you have to write a zone file for each web site.
* In the registrars’databases, your DNS server will be listed as the nameserver for those domain names.

/etc/named.conf is your directory listing of zone files

**Your Responsibility in DNS**

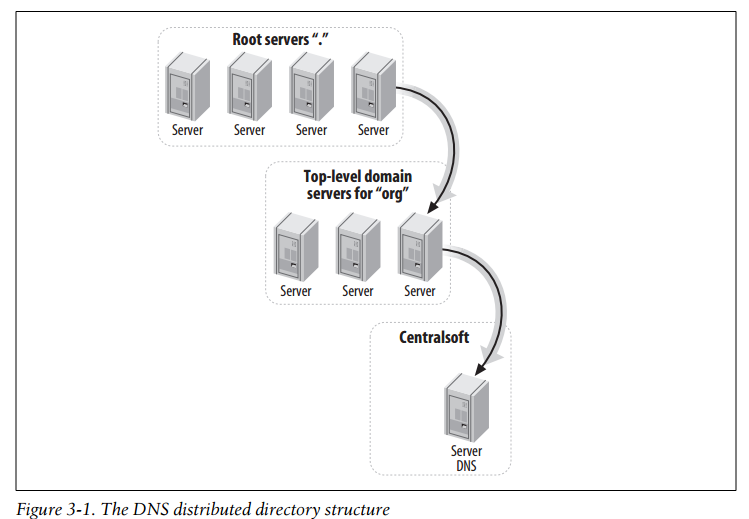
When you pay a fee and register a domain,You have to provide the names and addresses of two servers, and they have to be registered in the DNS system

**The Distributed Method of Resolving Domain Names**

The directory has three levels. The first group of servers is called root servers, because they provide the starting point for queries. The second group consists of the top-level domain servers. TLDs include .com, .net, .org, .mil, .gov, .edu, and so on, as well as country domains such as .de

Figure 3-1 depicts the DNS structure. At the top of the figure, you can see a representation of the Internet’s root servers. These servers contain only the names and IP addresses of the next level of servers and are responsible solely for redirecting requests to particular TLDs.

In the center of the figure, you see some of the servers for the .org TLD. These servers contain the names and IP addresses of all registered DNS servers with the suffix .org. If you register a domain with an .org suffix, its IP address will reside in each of .org’s TLD servers. You will have to provide the remaining information on any subdomains, including servers within your domain. The bottom layer in Figure 3-1 represents a primary nameserver called server1.centralsoft.org. It functions as the DNS server for a number of domains, as you’ll see later. For now, just know that server1.centralsoft.org represents the part of the DNS system that you will have to manage

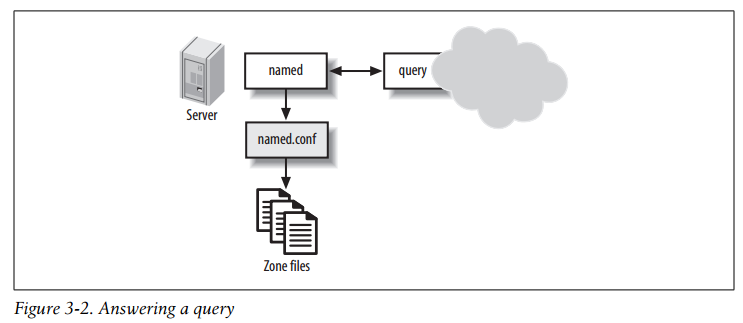


**Finding a Domain**

* BIND provides the mechanism for reading the directory. When your computer needs to find the address for a web site, it queries the DNS servers you specify
* say your browser wants to find www.google.com. BIND’s “client” executes a command that essentially asks its DNS server whether it knows the address of the web site. If the DNS server doesn’t know the address, it asks a root server for the address
* The root server replies, “I don’t know, but I do know where you can find the answer. Start with the TLD servers for .com.” And it provides the IP address of a server that knows all the domains (quite a lot!) that are registered directly under .com.
* The resolver on the DNS server then queries a .com server for the address. The .com server says, “I don’t have that information, but I know a nameserver that does. It has an address of 64.233.167.99 and its name is ns1.google.com.”
* resolv.conf controls the queries that browsers and other clients make for domain names, while named answers the queries and makes sure information is kept up-to-date on all servers.

**Answering Queries**

In the upper-left corner of the figure is a drawing of a server tower (in our example this server is called server1.centralsoft.org; it performs the same function as ns1. google.com). Assume the server is running Linux and BIND. A server at a higher level directs resolvers to the system (in the case of server1.centralsoft.org, a TLD nameserver for the .org domain sends the requests).



* The named daemon listens on UDP port 53 for anyone making requests for names in the domain. When named receives a request, it consults its configuration file, /etc/ named.conf.
* If the server has information on the domain in question, it looks in the appropriate zone file. If the zone file has the information requested, the server hands it off to the system querying for the information.
* Some people refer to configuration files as rule files. This makes some sense because correct DNS operation requires tight compliance with its rules and protocols

**Primary and Secondary DNS Servers**

Answering a query named Server query named.conf Zone files. The Domain Name System sider one server the primary or master server (where you will make all manual updates) and the other server the secondary or slave server. BIND then allows the secondary server to contact the primary one and automatically replicate the directory—a practice called a zone transfer.

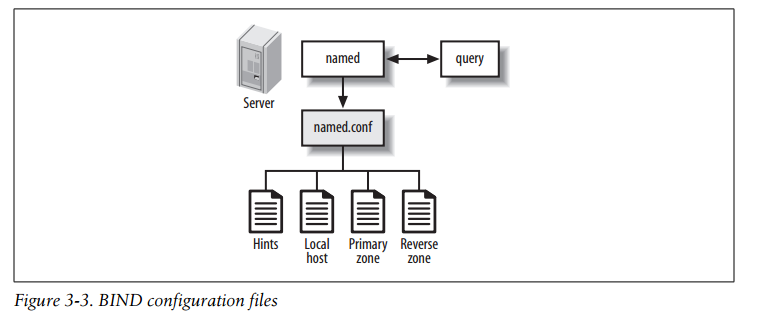
Secondary servers are authoritative, just as primary servers are. That is, secondary servers can respond to queries and give out information on all the zones for which they are responsible.

**Caching-Only Servers**

DNS offers caching-only servers. Administrators use these to reduce the load on authoritative servers. A caching server has no authority; it simply makes DNS work faster by storing the domain names it gets from authoritative servers and offering them to its clients.

**Editing the Configuration Files**

The named.conf file, which coordinates the whole system on each BIND server and points to the rest named.conf



When named receives a request it consults its own small directory, the named.conf configuration file

our sample named.conf. Comments follow double slash marks:

options {

pid-file "/var/run/bind/run/named.pid";

directory "/etc/bind";

// query-//

// a master nameserver config

//

zone "." {

type hint;

file "db.root";

};

zone "0.0.127.in-addr.arpa" {

type master;

file "db.local";

};

zone "158.253.70.in-addr.arpa" {

type master;

file "pri.158.253.70.in-addr.arpa";

};

zone "centralsoft.org" {

type master;

file "pri.centralsoft.org";

};

**The Primary Zone File**

The primary zone file contains the bulk of the configuration information DNS needs.The first lines provide the information needed to sync with your secondary or slave server(s):

@ IN SOA server1.centralsoft.org. root.localhost. (

2006012103; serial-no

28800; refresh, seconds

7200; retry, seconds

604800; expiry, seconds

86400 ); minimum-TTL, seconds;

This is an SOA record. SOA is the to Start of Authority, which distinguishes this as information for authoritative servers (both primary and secondary) as opposed to caching servers

* Name The root name of the zone.
* Class The DNS class. A number of classes exist, but the vast majority of sites use the IN (Internet) class.
* Type The type of DNS resource record. In this case, this is an SOA resource record.
* Email address The email address of the person responsible for the domain.
* The following lines in the SOA record contain fields for the slave server’s benefit:
* Serial-no The serial number for the current configuration.
* Refresh The interval at which a slave DNS server should check with the master to determine whether a zone transfer is required.
* Expiry The length of time for which a slave should try to contact the master before expiring the data it contains
* Minimum-TTL The default time-to-live for this domain in seconds.

**The Reverse Zone File**

A reverse zone file maps IP addresses to names. It looks almost like a mirror of the primary zone file; instead of listing the names first, the reverse zone file lists the IP addresses first

**BIND Tools**

BIND comes in three pieces: the named daemon, the resolver library, and some tools.

One tool you have already used is dig, which administrators use to interrogate DNS nameservers. dig does a DNS lookup and displays both the answers returned from the nameservers and statistics about the query.

**nslookup**

nslookup works similarly to dig but is deprecated in Linux

nslookup queries Internet domain nameservers in two modes: interactive and noninteractive. The interactive mode allows you to query nameservers for information about various hosts and domains, or to print a list of hosts in a domain.

# nslookup ns1.google.com

Server: 68.94.156.1

Address: 68.94.156.1#53

**rndc**

BIND provides the rndc command as part of the installation. rndc allows you to administer named using the command line. The utility sends the commands given onthe command line to the running named server, which acts on them. rndc is also used by the BIND 9 system initialization script.

The rndc command takes the following form:

# rndc rndc-options command command-options

**VIRTUALIZATION**

**Virtualization** means to create a virtual version of a device or resource, such as a server, storage device, network or even an operating system where the framework divides the resource into one or more execution environments.

**Linux virtualization** refers to running one or more virtual machines on a physical computer that's operated by the Linux open source operating system. Linux virtualization can be used for isolating specific apps, programming code or even an operating system itself, as well as for security and performance testing purposes

Linux virtualization lies at the heart of today’s trends in data center consolidation, high-performance computing, rapid provisioning, business continuity, and workload management.

T he different guests are isolated from each other much more than processes are isolated within a single operating system. This isolation provides security and robustness, because a failure or compromise in one guest doesn’t affect the others. The virtualization layer performs many functions of an operating system, managing access to processor time, devices, and memory for each guest

**Installing Xen on Fedora 5**

The Xen Project hypervisor is an open-source hypervisor(A hypervisor or virtual machine monitor (VMM) is a piece of computer software, firmware or hardware that creates and runs virtual machines.), which makes it possible to run many instances of an operating system or indeed different operating systems in parallel on a single machine (or host). It is used as the basis for a number of different commercial and open source applications, such as: server virtualization, Infrastructure as a Service (IaaS), desktop virtualization, security applications, embedded and hardware appliances. The Xen Project hypervisor is powering the largest clouds in production today

In this we’ll see how to install Xen on a single machine to manage two operating systems. As Xen makes its way into the standard Linux distributions, installation will become smoother. But for now, some manual labor is

needed. We’re using Fedora Core 5 (FC5) as the Xen host operating system, since it supports Xen 3.0 out of the box. Let’s ask yum (a package manager similar to Debian’s apt-get or Red Hat’s up2date) about Xen:

# yum info xen

Loading "installonlyn" plugin

Setting up repositories

core [1/3]

updates [2/3]

extras [3/3]

Reading repository metadata in from local files

Available Packages

Name : xen

Arch : i386

Version: 3.0.2

Release: 3.FC5

Size : 1.4 M

Repo : updates

Summary: Xen is a virtual machine monitor

**Description:**

This package contains the Xen hypervisor and Xen tools, needed to run virtual machines on x86 systems, together with the kernel-xen\* packages. Information on how to use Xen can be found at the Xen

project pages.

Virtualisation can be used to run multiple versions or multiple Linux distributions on one system, or to test untrusted applications in a sandboxed environment. Note that the Xen technology is still in development, and this RPM has received extremely little testing.

Run the system-config-securitylevel program or edit /etc/selinux/config to looks as follows:

# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - SELinux is fully disabled.

SELINUX=Disabled

# SELINUXTYPE= type of policy in use. Possible values are:

# targeted - Only targeted network daemons are protected.

# strict - Full SELinux protection.

SELINUXTYPE=targeted

If you changed the SELINUX value from enforcing, you’ll need to reboot Fedora before proceeding.

This command will install the Xen hypervisor, a Xen-modified Fedora kernel called domain 0, and various utilities:

# yum install kernel-xen0

Now you can reboot. Xen should start automatically, but let’s check:

# /usr/sbin/xm list

Name ID Mem(MiB) VCPUs State Time(s)

Domain-0 0 880 1 r----- 20.5

The output should show that Domain-0 is running. Domain 0 controls all the guest operating systems that run on the processor, similarly to how the kernel controls processes in an operating system.

**Installing a Xen Guest OS**

Xen does not start the guest operating system automatically. You need to type this command on the host:

# xm create guest1

A minimal guest configuration file looks something like this:

Installing Xen on Fedora 5 | 203

1. A unique guest domain name:

name="vm01"

2. A Xen-enabled kernel image pathname for the guest domain:

kernel="/boot/vmlinuz-2.6.12.6-xenU"

3. A root device for the guest domain:

root="/dev/hda1"

4. Initial memory allocation for the guest, in megabytes:

memory=128

5. The disk space for the guest domain. This is defined in one or more disk block device stanzas, each enclosed in single or double quotes:

disk = [ 'stanza1', 'stanza2' ]

A stanza consists of a string of three parameters ('host\_dev, guest\_dev, mode').host\_dev is the domain’s storage area as seen by the host.

6. Network interface information in a vif directive. This directive may contain a stanza for each network device. The default network is specified with:

vif=[ '' ]

A dhcp directive controls whether DHCP is used or the interface information is hard-coded. The following specifies the use of DHCP:

dhcp="dhcp"

Once you have a guest configuration file, create the Xen guest with this command:

# xm create -c guest\_name

where guest\_name can be a full pathname or a relative filename

**Installing VMware**

* VMware is a virtualization and cloud computing software provider for x86-compatible computers.
* VMware has also submitted its own open source contributions to the kernel developers, realizing that VMware will run better on Linux if VMware gives the Linux kernel a little help
* We downloaded Vmware-server-1.0.1-29996.tar.gz and decompressed it to an installation directory called vmware-server-distrib. Inside the directory we found vmwareinstall.pl and ran it with the command ./vmware-install.pl.
* The installation program began and displayed the following messages:

Creating a new installer database using the tar3 format.

Installing the content of the package.

In which directory do you want to install the binary files?[/usr/bin]

The installation script will also ask you to run the configuration command:

Before running VMware Server for the first time, you need to configure it by invoking the following command: "/usr/bin/vmware-config.pl". Do you want this program to invoke the command for you now? [yes]

As the installation process ends, you will see the following messages:

Starting VMware services:

Virtual machine monitor done

Virtual Ethernet done

Bridged networking on /dev/vmnet0 done

Host-only networking on /dev/vmnet1 (background) done

Host-only networking on /dev/vmnet8 (background) done

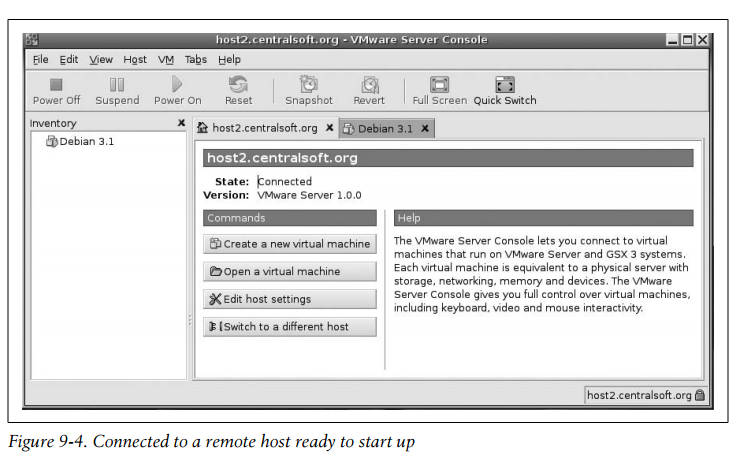
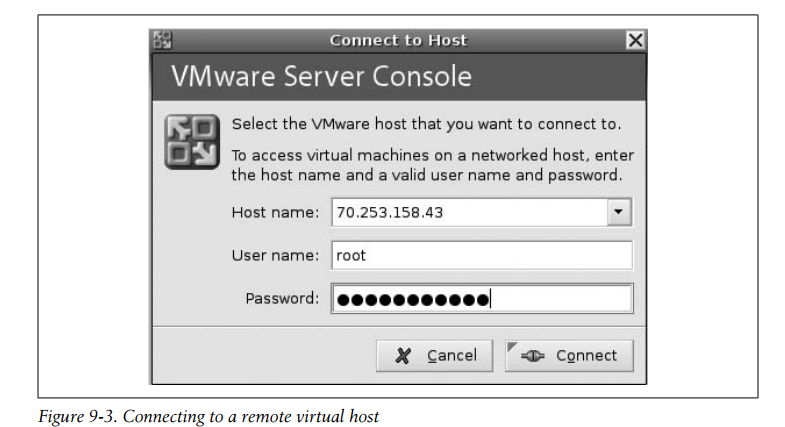
NAT service on /dev/vmnet8 done

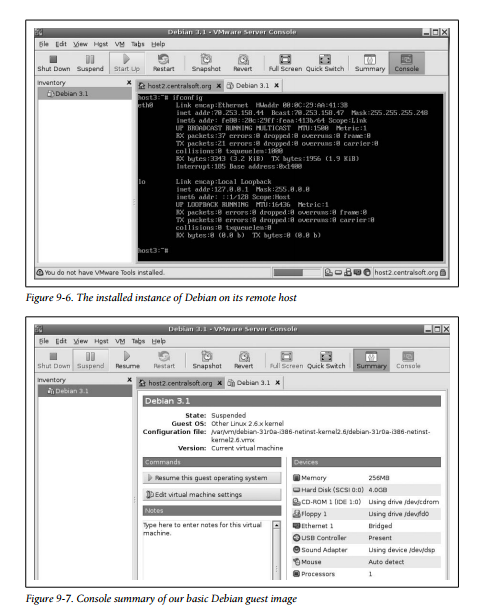
Starting VMware virtual machines done

We ran the command:

$ gksu vmware-server-console

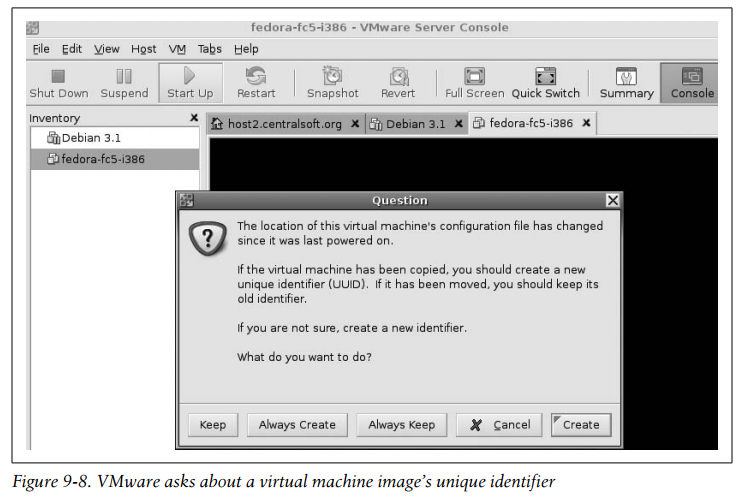
We then configured the console to connect to our guest operating system remotely.





**Installing a VMware Guest OS**

We downloaded Fedora Core 5 from VMware’s community site, moved it to the Virtual Machines directory, and decompressed it as we did with Debian



VMware’s management console noticed we added an image. In order to distinguish between possible multiple images, it prompted us for a unique identifier (UUID) in the dialog shown in Figure 9-8. Because we copied Fedora 5 and have all the files making up the image, it did not matter which option we chose from the dialog. When you open a new virtual machine, VMware gives you a chance to verify the virtual hardware configuration. Figure 9-9 gives you an idea of the virtual hardware inventory available for Fedora Core 5. In addition to downloading images and loading them into the management console, you can install a Linux operating system from a standard Linux distribution’s CD-ROM.

