Linux COE

TATA CONSULTANCY SERVICES

ABOUT THIS DOCUMENT

Document Name: c-Perpetual in Linux
Version No.: 1.0

Purpose:

This document is intended to move one from E0 to E2 level in Linux Administration just in 55 days

Context:

1. Basic Computer Hardware architecture

1.1. POST

Important components of computers are CPU, RAM and I/O Devices. So when the input is given via Input Device, it first goes to RAM, then CPU processes the request and output again goes to RAM. This output will be given to us via output device.

To understand how computers are working, we should start with Power On. When the system is powered on, the BIOS will perform Self Test to check if the minimum hardware (important) of hardware required for proper working of system is alright. Important components of computers are CPU, RAM and I/O Devices. So when the input is given via Input Device, it first goes to RAM; CPU processes the request from RAM and output again goes to RAM. This output will be given to us via output device. This is commonly referred as POST (Power-On Self Test).

POST is diagnostic software available in BIOS ROM, exactly on the address which is generated by processor on reset. The Sequence hardware tested on Power On Self Test is as below:

Main Processor
Numeric Processor
BIOS ROM
Base Memory Size
Ext. Memory Size
Floppy Drive A
Floppy Drive B
Hard disk controller
Display Type
Serial Port
Parallel Port
Cache memory information
PCI cards information

Once POST completed, the control goes to Boot Strap Loader, which reads the content of Master Boot Record (MBR). MBR is nothing but (Track 0, sector 1) first sector of the first hard drive There on it finds location of Operating system to be loaded and loads the same, which is called booting. While booting, it loads BIOS (driver software for the Basic I/O devices. For all other advanced I/O devices, the driver software may be in built with operating system or has be installed manually.

Once booting is completed, control given to user, so that users can execute commands or do anything they want to be computed.

1.2. Keyboard Functionality

Keyboard is a standard Input device, where buttons or keys are arranged in matrix format. When no key is pressed, all the points in matrix will be open circuit.

When a key is pressed, a pair of conductive lines on the circuit below will be (touched) closed. This bridges the gap between conductive lines in a keyboard circuit and allows electric current to flow (the open circuit is closed).

Now microprocessor in the keyboard circuit will send scanning signal to the computer. Whenever the key is pressed, the microprocessor in the keyboard generates a "make code" corresponding to the key.

The generated code is sent to the computer either via a keyboard cable (using on-off electrical pulses to represent bits) or over a wireless connection. It may be repeated.

A keyboard controller inside the computer receives the signal bits and decodes them into the ASCII code (appropriate to the keypress). With the help of keyboard driver, keyboard controller then decides if the key pressed is a display character (to be displayed on the screen) or a control character (to do some action).

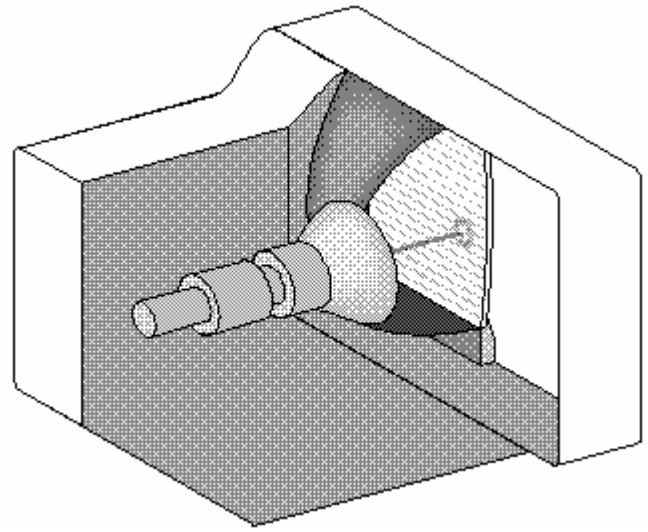
For example, if the key pressed is a display character, then it will be sent to video memory, so that on the next refresh, the video controller displays this character on a screen. If it is a control character, appropriate action will be taken place. For example, if you press print screen, keyboard controller gives information to the CPU, then CPU instructs printer controller that the current content of video memory should be printed to the default printer. The printer controller will take care of rest of the sctions.

When the key is released, a break code (different than the make code) is sent to indicate the key is no longer pressed. If the break code is missed (e.g. due to a keyboard switch) it is possible for the keyboard controller to believe the key is still pressed down.

Certain key presses are special, namely Ctrl-Alt-Delete and SysRq, but what makes them special is a function of software. In the PC architecture, the keyboard controller (the component in the computer that receives the make and break codes) sends the computer's CPU a hardware interrupt whenever a key is pressed or released. The CPU's interrupt routine which handles these interrupts usually just places the key's code in a queue, to be handled later by other code when it gets around to it, then returns to whatever the computer was doing before. The special keys cause

the interrupt routine to take a different "emergency" ex it instead. This more trusted route is much harder to intercept.

1.3. Monitor Functionality



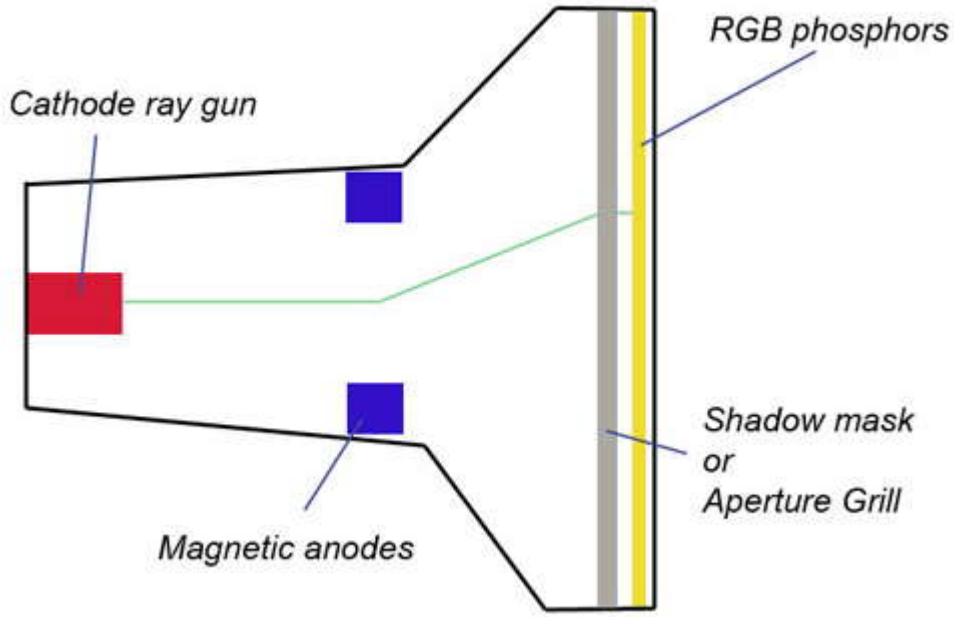
The picture or letters or whatever that appears on your monitor comes from the graphics card in your computer. Graphics card gets data from video buffer, process the same and renders the data suitable for the monitor to display (called dot informations). A wired output runs from the graphics card to the monitor.

Both the graphics card and monitor adhere to the same set of specifications, so that they can communicate properly. The standards are set out by VESA, which defines things like how monitors identify themselves to the computer.

There are two different types of monitor, one is CRT monitor and other is LCD monitor. CRT stands for Cathode Ray Tube, and is descriptive of the technology inside the monitor, which is the similar technology used by television. CRTs receive their picture through an analogue cable, and that signal is decoded by the display controller, which handles the internal components of the monitor - think of it as the mini-CPU for the monitor.

CRTs have a funnel shape and the sharper edge funnel has an electron gun or cathode. The electron gun fires electrons towards the front portion of the monitor and the passage is vacuum. The electrons fired forward are called Cathode Rays. These rays correspond to the red, green and blue channels of the display and video card.

The anode will be there tt the neck of the funnel-shaped monitor, which is magnetized according to instructions from the graphics controller. As electrons pass the anode, they are shunted or pulled in one direction or the other depending on how magnetic the anode is at that time. This moves the electrons towards the correct part of the screen.

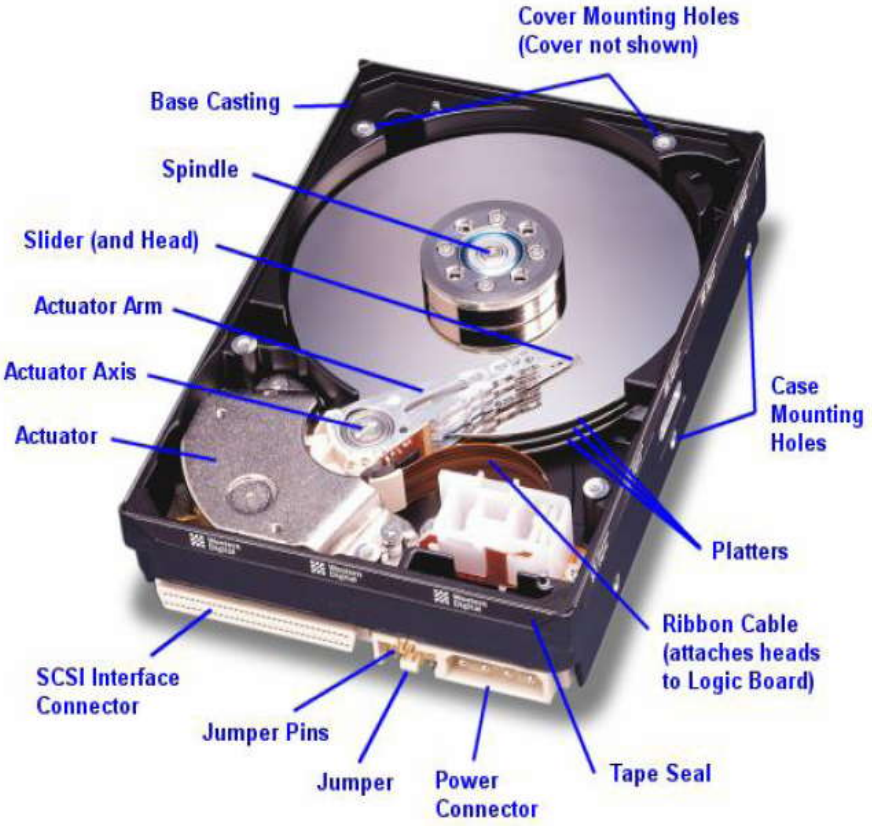


The electrons pass through a mesh, and this mesh defines the individual pixels and resolution on the screen. Electrons that pass through the mesh then hit the phosphor coating which is on the inside of the glass screen. When the particles hit the phosphor, they immediately light up - causing the light to shine through the front of the monitor, thus making up the picture on the screen.

Horizontal and vertical deflection plates are there which makes horizontal and vertical scanning. When the scanning spots the point, the anode allows the cathode rays to pass and reach the phosphors coating or not based on the instruction from graphics controller. If that point needs to be illuminated, then anode allows and accelerates cathode ray to reach phosphors coating. If not, then anode will not allow cathode ray to pass through it.

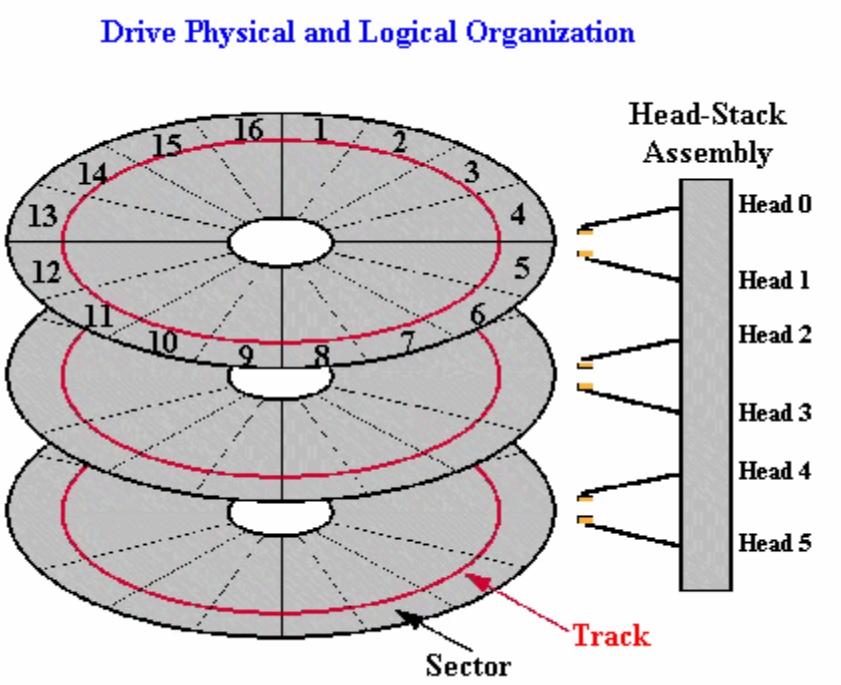
1.4. Hard disk drive Functionality

Hard Disk drives record data by magnetizing ferromagnetic material directionally, to represent either a 0 or a 1 binary digit. They read the data back by detecting the magnetization of the material.



A hard disk drive has spindle which holds one or more flat circular disks called platters, onto which the data is recorded. The platters rotate at high speed, driven by a special spindle motor connected to the spindle. Special electromagnetic read/write devices called heads are mounted onto sliders and used to either record information onto the disk or read information from it. The sliders are mounted onto arms, all of which are mechanically connected into a single assembly and positioned over the surface of the disk by a device called an actuator. A logic board controls the activity of the other components and communicates with the rest of the PC. Corresponding tracks in all platters called cylinder. The platters are made from a non-magnetic material, and are coated with a thin layer of magnetic material. Each platter has two heads, one on the top of the platter and one on the bottom, so a hard disk with three platters (normally) has

six surfaces and six total heads.



These platters are logically divided in to number of concentric circles called tracks. Tracks are again divided logically into number of sectors, where data will be written.

1.5. How Data stored in a Hard disk

There are two methods for data addressing: CHS (cylinder-head-sector) and LBA (logical block address). CHS is used on most IDE drives, while LBA is used on SCSI and enh anced IDE drives. CHS addresses data by simply specifying the cylinder (radius), head (platter side), and sector (angular position). LBA assigns each sector of the drive a sequential number, which is simpler.

Sectors are not read individually on most PCs; they are grouped together into continuous chunks called clusters. A typical job, such as loading a file into a spreadsheet program, can involve thousands or even millions of individual disk accesses, and loading a 20 MB file 512 bytes at a time would be rather inefficient:

The first step in accessing (Read or write) the disk is to figure out where on the disk to look for the needed information. This is normally expressed in terms of the cylinder, head and sector of the drive, the system want to access (read or write). A request is sent to the drive over the disk drive interface giving it this address and asking for the sector to be read.

The hard disk's control program first checks to see if the information requested is already in the hard disk's own internal buffer (or cache). If it available, then the data will be provided from the buffer itself.

As cylinder number tells the disk which track to look at on the surface of the disk, the controller board performs any necessary action to instructs the actuator to move the read/write heads to the appropriate track.

When the heads are in the correct position, the controller activates the head specified in the correct read location. The head begins reading the track looking for the sector that was asked for. It waits for the disk to rotate the correct sector number under itself, and then reads the contents of the sector.

The controller board coordinates the flow of information from the hard disk into a temporary storage area (buffer). It then sends the information over the hard disk interface, usually to the RAM.

1.6. What happens while reading a file?

Consider reading a file from FAT file system. While you execute dos command “type filename”, how the file is retrieved from hard disk and displayed on a screen?

When the command is typed to read a file, system will look at the Root Directory Table (RDT) of FAT file system and checks if there is such file available. If yes, it will find out the track number and starting sector (in which file is stored) of File Allocation Table (FAT). Then control moves to File Allocation Table, and collects the complete list of sectors the file is stored in sequence.

Once the details of all sectors (in sequence) and track number are collected, then Hard disk controller instructs stepper motor to bring Read/Write head to the appropriate cylinder. Then it waits for the disk to rotate till the correct sector number comes under the Read/Write head, and then reads the contents of the sector. After reading the starting sector, it again waits till the next sector comes under it, and then reads the data. The same action happens till it find End of File.

Immediately after the Read /Write head reads the content of sector, the bit information is send to system memory via hard disk drive interface. As we requested to read a file, CPU instructs video controller to display this content on screen.

Video controller reads RAM and converts the ACSII data to printable dot information and saves the same in video buffer. On next refresh, the data will be displayed on the screen.

While writing data, platters rotates to locate the appropriate sector; read write coil moves to locate the track; then hard disk controller passes current (appropriate to the data) on to the Read write coil, so that magnetic flux produced on the appropriate sector.

1.7. What happens while editing a file and saving it in hard disk?

When the command is typed to edit a file, the same action happens as reading file, but the file is open to edit. For example if you are adding few more line to the existing file, what happens?

While you are typing content in the file, what ever data you type will first be saved in RAM. When you click to save it onto the file, hard disk controller locates the unused or empty sectors. Then it instructs motor to bring the Read / Write coil to appropriate track. As a platter rotates, read/ Write head waits till appropriate sector comes under it, and then start writing. Writing is nothing but, disk controller passes current (appropriate to the data) on to the Read write coil, so that magnetic flux produced on the appropriate sector. The direction of magnetic flux produced is appropriate to the data written.

1.8. What happens when you print a file?

When the command is typed (click print) to print a file, the same action happens as reading file, then CPU sends information to the printer controller that there is a file to print and then it sends data to be printed to printer buffer.

Printer controller finds out if the printer is ready to take data, if yes then it converts the data to the printer through printer interface cable. Once data is received by the printer, printer electronics available inside the printer converts the data into printable format and prints the same by activating the mechanical assembly accordingly. Once the data is printed, printer electronics send information to the printer controller that the file has been printed. The same information is send to CPU from printer controller to display it on the monitor.

2. Tasks of Linux System Administrator

1. Tasks of Linux System Administrators

As an administrator, one should know how to perform the following tasks:

- Installing operating system
  - Different types of installations like installing from CDROM, Hard Drive, network installation such as NFS, FTP, HTTP etc.. and performing unattended installation like Kick start
- Startup and shutdown concepts
  - Understanding what happens while startup Linux system, different run levels, creating bootable disks, troubleshooting the booting issues, commands to boot the system in different run levels, commands to shutdown or reboot the system, making a script to run at boot time, etc…
- User and Group account administration
  - Understanding different types of user accounts and group accounts, creating user accounts, modifying user parameters, setting up password, setting up password policy, deleting user accounts, creating group accounts, managing group accounts, deleting group accounts, adding users to groups, etc…
- Device Management and file system administration
  - Configuring new devices to the system, adding new hard disk, creating partitions, understanding different file systems supported by Linux, creating file system, mounting file system, repairing file system, mounting removable medias like floppy and CD, mounting USB drives, burning data to CD, etc…
- Software installation
  - Installing a third party software, installing software in rpm format using rpm command, installing software using yum, installing software from source code, upgrading and uninstalling packages, etc…
- Managing the processes
  - Managing all running processes, pushing jobs to background, pulling jobs from background to foreground, understating /proc file system and modifying the running kernel parameters, etc…
- Performance Tuning
  - Understanding tools available for measuring the system performance, Managing swap space, etc…
- Printer Administration
  - Installing printer in Linux, sharing printer through samba, administering print jobs, Administering printer, etc…

- X Window configuration
  - Installing and configuring X Server and X client, Installing and configuring VNC Server and client, etc…

As a network Administrator, one should know to perform the following:

- TCP/IP and Network configuration
  - Understanding TCP/IP protocol suite, Understanding IP Addressing and class A, B, C, D, and D IP address ranges, default subnet mask, Configuring Ethernet Card, understanding IP Forwarding, subnet mask, Default Gateway, MAC address, ARP and RARP, Assigning virtual IP addresses to Ethernet card, Understanding Basic Networking
- Setting up NFS
  - Installing and configuring NFS, Understanding Advantages/disadvantages of NFS
- Setting up Samba
  - Installing and configuring Samba, Understanding Advantages/disadvantages of Samba
- Setting up NIS, NIS+ and LDAP
  - Installing and configuring NIS as an authentication server, Installing and configuring NIS+ as an authentication server, Installing and configuring LDAP as an authentication server, Advantages/disadvantages of NIS, NIS+ and LDAP
- Setting up DNS and DHCP
  - Installing and configuring DNS, Installing and configuring DHCP, Configuring DNS clients via DHCP
- Setting up Web Server
  - Installing and configuring http and Apache Web server, Concepts of virtual hosts and steps to configure the same
- Setting Mail server
  - Installing and configuring send mail server, Concepts of virtual servers and virtual users
- Setting up proxy
  - Understanding the concept of proxy server, Configuring squid proxy
- Setting up firewall
  - Understanding the concept of firewall, Understanding all possible tasks that can be done through iptables
- Setting up Routing
  - Setting up Linux System as a router
- Performing Shell scripts to ease the system administration

2. About TCP/IP Addressing

IP Addressing:

An IP address in IP Version 4 is a 32-bit number that uniquely identifies a host (computer or other device, such as a printer or router) on a TCP/IP network.

IP addresses are normally expressed in dotted-decimal format, with four numbers separated by periods, such as 192.168.123.132. To understand how subnet masks are used to distinguish between hosts, networks, and subnetworks, examine an IP address in binary notation.

For example, the dotted-decimal IP address 192.168.123.132 is (in binary notation) the 32 bit number 11000000010100011110111000100. This number may be hard to make sense of, so divide it into four parts of eight binary digits.

These eight bit sections are known as octets. The example IP address, then, becomes 11000000.10101000.01111011.10000100. This number only makes a little more sense, so for most uses, convert the binary address into dotted-decimal format (192.168.123.132). The decimal numbers separated by periods are the octets converted from binary to decimal notation.

The first part of an IP address is used as a network address, the last part as a host address. If you take the example 192.168.123.132 and divide it into these two parts you get the following:

```
192.168.123. Network .132 Host
```

-or-

```
192.168.123.0 - network address. 0.0.0.132 - host address.
```

Subnet mask

The second item, which is required for TCP/IP to work, is the subnet mask. The subnet mask is used by the TCP/IP protocol to determine whether a host is on the local subnet or on a remote network.

In TCP/IP, the parts of the IP address that are used as the network and host addresses are not fixed, so the network and host addresses above cannot be determined unless you have more information. This information is supplied in another 32-bit number called a subnet mask. In this example, the subnet mask is 255.255.255.0. It is not obvious what this number means unless you know that 255 in binary notation equals 11111111; so, the subnet mask is:

```
11111111.11111111.11111111.0000000
```

Lining up the IP address and the subnet mask together, the network and host portions of the address can be separated:

```
11000000.10101000.01111011.10000100 -- IP address (192.168.123.132)
11111111.11111111.11111111.00000000 -- Subnet mask (255.255.255.0)
```

The first 24 bits (the number of ones in the subnet mask) are identified as the network address, with the last 8 bits (the number of remaining zeros in the subnet mask) identified as the host address. This gives you the following:

```
11000000.10101000.01111011.00000000 -- Network address (192.168.123.0)
00000000.00000000.00000000.10000100 -- Host address (000.000.000.132)
```

So now you know, for this example using a 255.255.255.0 subnet mask, that the network ID is 192.168.123.0, and the host address is 0.0.0.132. When a packet arrives on the 192.168.123.0 subnet (from the local subnet or a remote network), and it has a destination address of 192.168.123.132, your computer will receive it from the network and process it.

Almost all decimal subnet masks convert to binary numbers that are all ones on the left and all zeros on the right. Some other common subnet masks are:

```
Decimal Binary 255.255.255.192 1111111.11111111.1111111.11000000 255.255.255.224
1111111.11111111.1111111.11100000
```

IP Classes:

Internet addresses are allocated by the InterNIC (http://www.internic.net), the organization that administers the Internet. These IP addresses are divided into classes. The most common of these are classes A, B, and C. Classes D and E exist, but are not generally used by end users. Each of the address classes has a different default subnet mask. You can identify the class of an IP address by looking at its first octet. Following are the ranges of Class A, B, and C Internet addresses, each with an example address:

- Class A networks use a default subnet mask of 255.0.0.0 and have 0-127 as their first octet. The address 10.52.36.11 is a class A address. Its first octet is 10, which is between 1 and 126, inclusive.

- Class B networks use a default subnet mask of 255.255.0.0 and have 128-191 as their first octet. The address 172.16.52.63 is a class B address. Its first octet is 172, which is between 128 and 191, inclusive.

- Class C networks use a default subnet mask of 255.255.255.0 and have 192-223 as their first octet. The address 192.168.123.132 is a class C address. Its first octet is 192, which is between 192 and 223, inclusive.

In some scenarios, the default subnet mask values do not fit the needs of the organization, because of the physical topology of the network, or because the numbers of networks (or hosts) do not fit within the default subnet mask restrictions. The next section explains how networks can be divided using subnet masks.

2. About Basic Networking

Networking is nothing but linking computing devices together with hardware and software that supports data communications across these devices.

Each computer in a (TCP/IP) network should be assigned with the IP Address. However, ultimate communication can not be achieved with the IP Address, because it is a logical address, not a permanent one. Therefore, IP address should be resolved into MAC Address or Hardware address, which is nothing but the port address of the Ethernet card, which is unique for each and every Ethernet card, and it is assigned by the manufacturer itself. Hence, U ltimate communication in a network is established with the MAC address.

As it is difficult to remember the IP address of each and every system, we are assigning hostname for each computer in a network. Hostname is resolved into IP address by Domain Name service (DNS). Then IP address will be resolved into MAC address by Address resolution protocol (ARP), which finally makes network communication.

Complete processes happening during data communication is defined with in 7 Layers, Which is called OSI (Open System Interconnect) layers.

Here are the 7Layers.

	OSI Model				
	Data unit	**Layer**	**Function**	**Protocols**	**Device Used**
Host layers	Data	7.Application	Network process to application File transfers, e-mail, Telnet and FTP, etc	DNS; FTP; TFTP; BOOTP, TELNET; etc…	Gateway
		6. Presentation	Data representation and encryption		Gateway Redirector
		5. Session	Establishes, manages and terminates connections between applications.	NetBIOS Names Pipes Mail Slots RPC	Gateway
	Segment / Datagram	4.Transport	End-to-end connections and reliability (TCP)	TCP, ARP, RARP;	Gateway advanced Cable Tester, Brouter
Media layers	Packet	3.Network	Path determination and logical addressing (IP)	ARP; RARP, ICMP; RIP; OSFP; IGMP;	Brouter ,Router, Frame Relay Device
	Frame	2.Data link	Physical addressing (MAC & LLC)	**Logical Link Control** ,802.1 OSI Model (LLC),802.2 Logical Link Control **Media Access Control** 802.3 CSMA/CD (Ethernet), 802.5 Token Ring	Switch ,Bridge, ISDN Router, Intelligent Hub ,NIC, Advanced Cable Tester
	Bit	1.Physical	Media, signal and binary transmission	IEEE 802 ,ISDN	Modem ,repeater Hubs ,Amplifier

3. Red Hat Enterprise Linux 4.0 Installation

1. Types of Installation in RedHat

- CD-ROM Graphical Installation
- NFS Installation
- FTP Installation
- HTTP Installation
- Hard Drive Installation
- Kick Start Installation

1.1 CD-ROM Graphical Installation

Boot the system from first CD

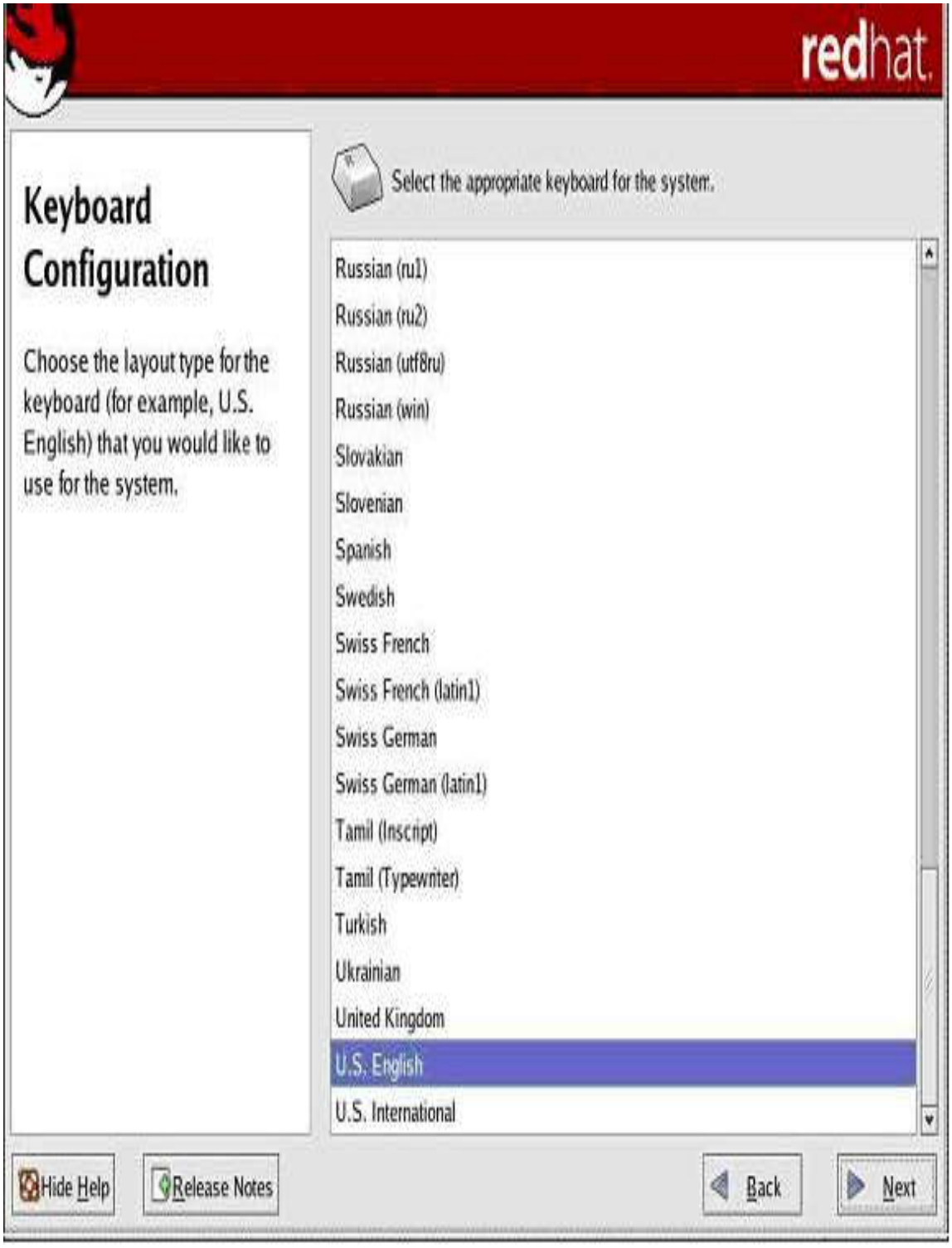
At the first screen press <enter> for the graphical install.

Press OK to test the media or SKIP to proceed.

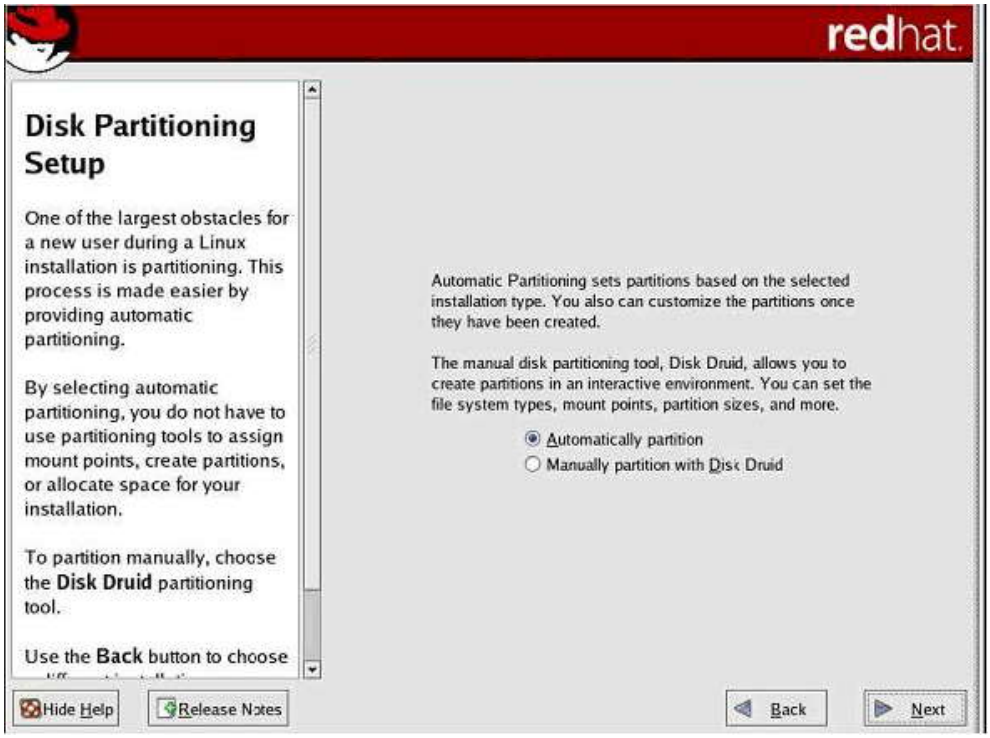
Begin the installation by clicking Next

Choose your language and click

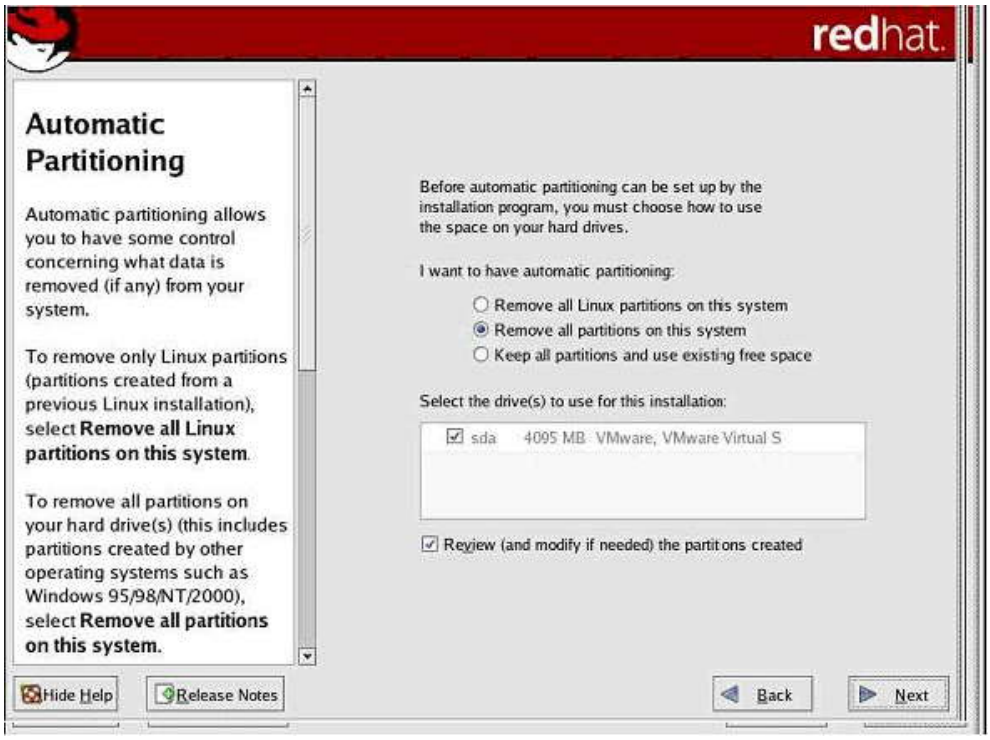
Next Choose your keyboard layout and click Next



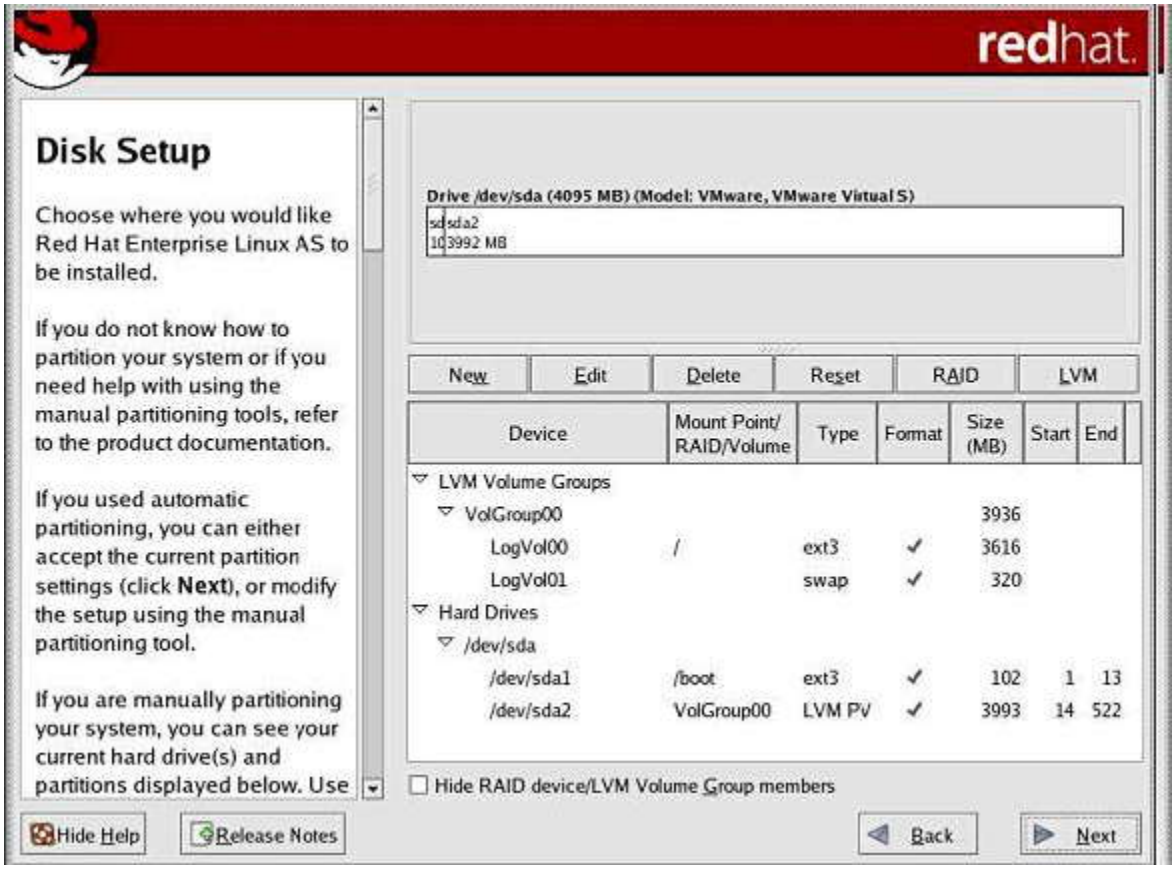
On this screen, you can choose to perform automatic partitioning, or manual partitioning using Disk Druid. Select Automatically Partition



Choose how you want automatic partitioning to use the space on the hard drive.

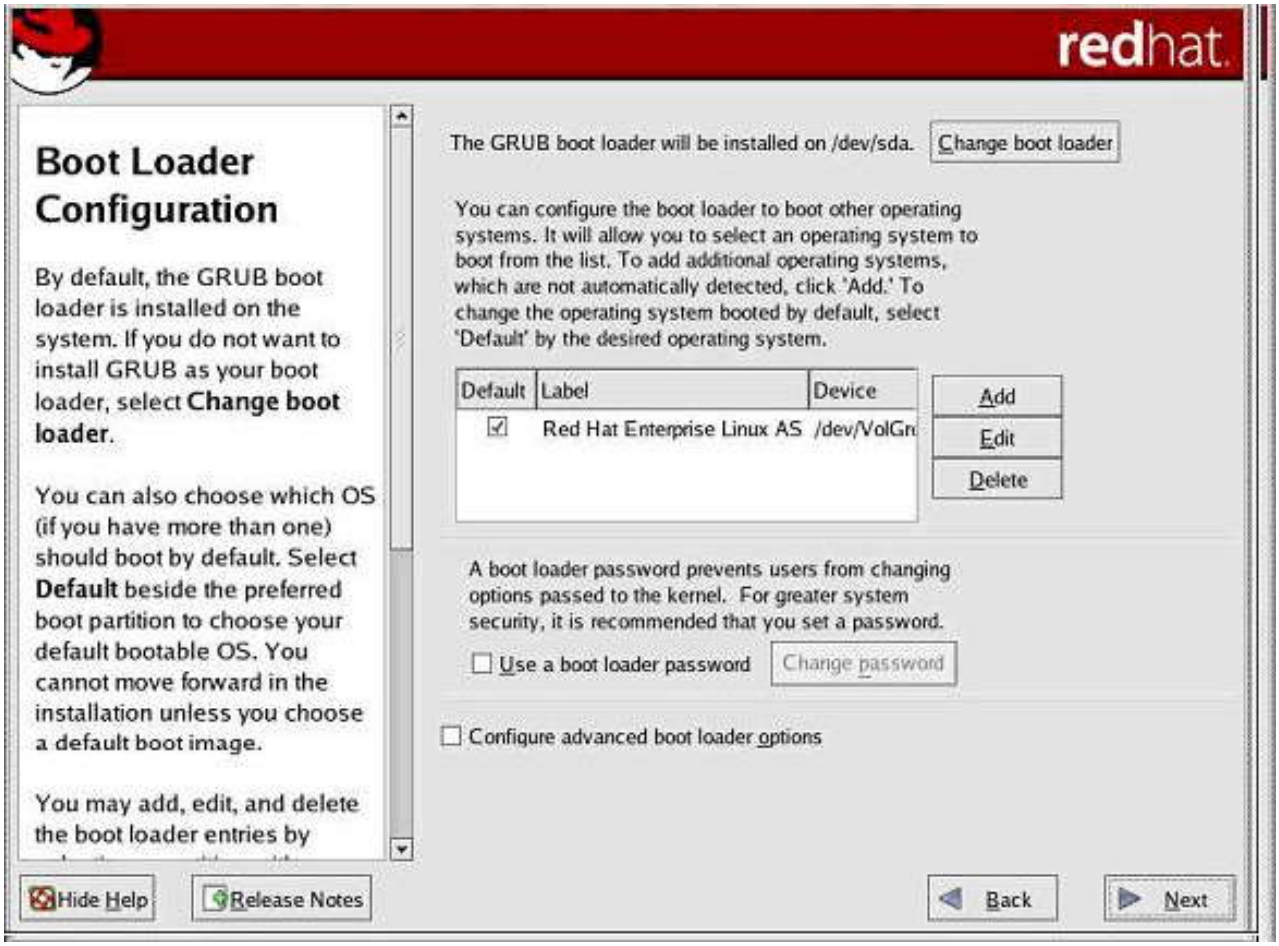


Confirm the disk set up is correct and click Next. The disk size information shown here may be different from what is displayed for your system.

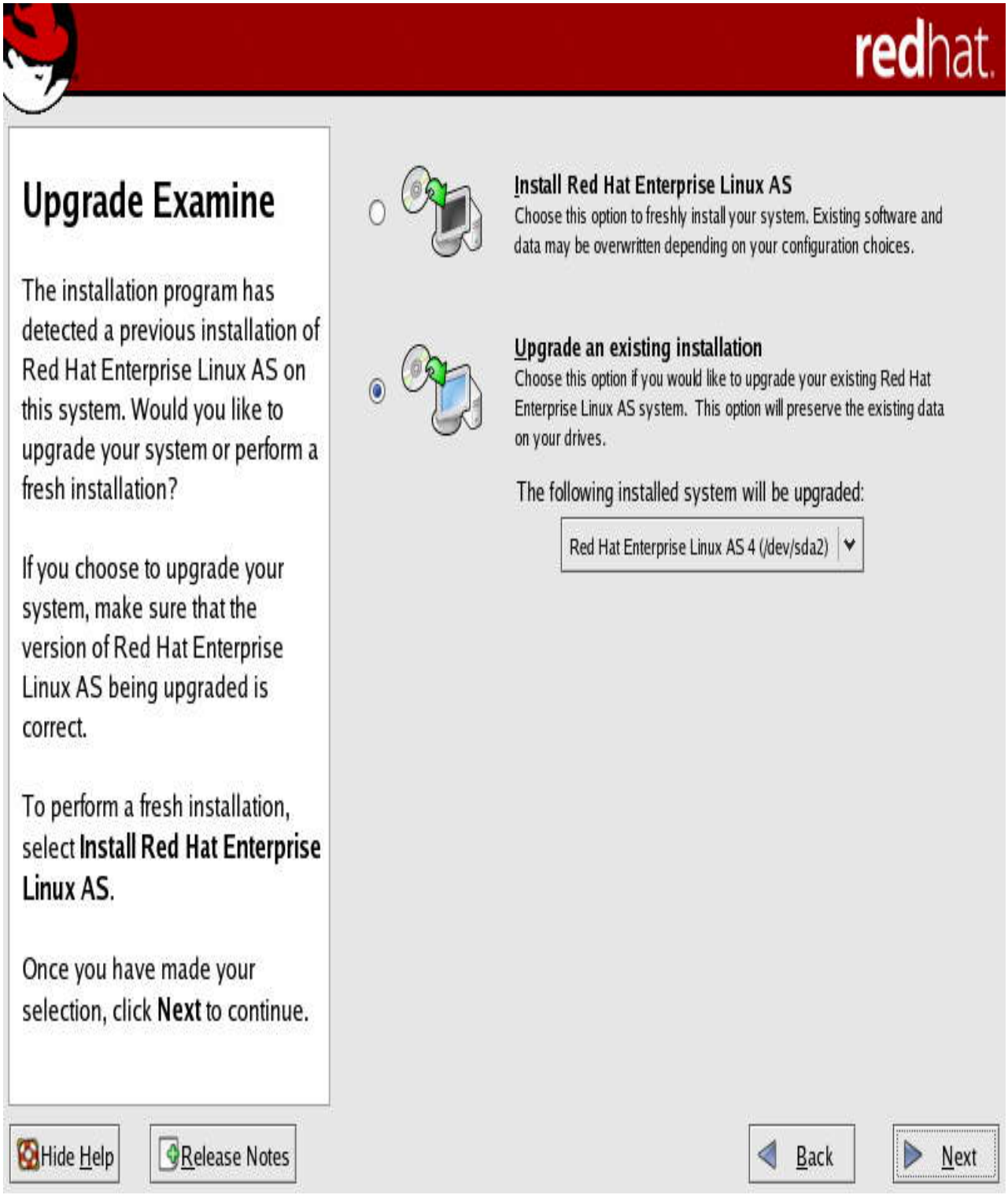


If you chose to partition manually, you must tell the installation program where to install Red Hat Enterprise Linux. This is done by defining mount points for one or more disk partitions in which Red Hat Enterprise Linux is installed. You may also need to create and/or delete partitions at this time.

Select a boot loader. GRUB (GRand Unified Bootloader), which is installed by default, is a very powerful boot loader. GRUB can load a variety of free operating systems, as well as proprietary operating systems with chain-loading (the mechanism for loading unsupported operating systems, such as DOS or Windows, by loading another boot loader).



If you are upgrading the OS you will get the following screen

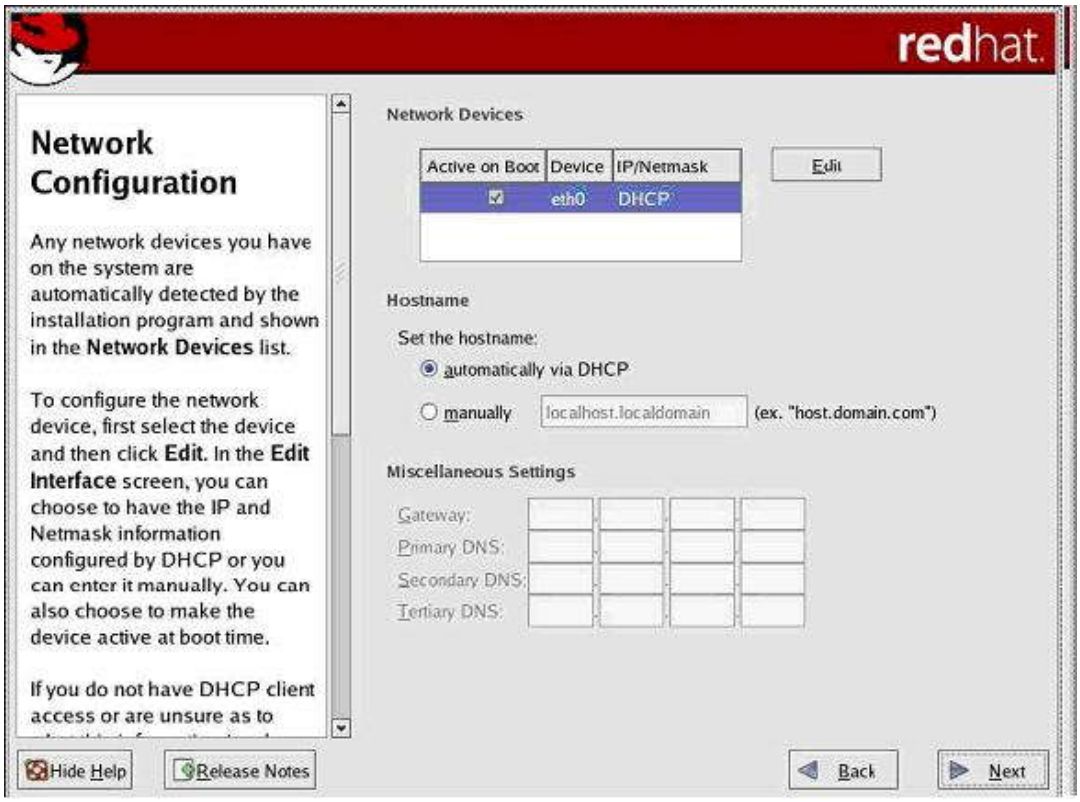


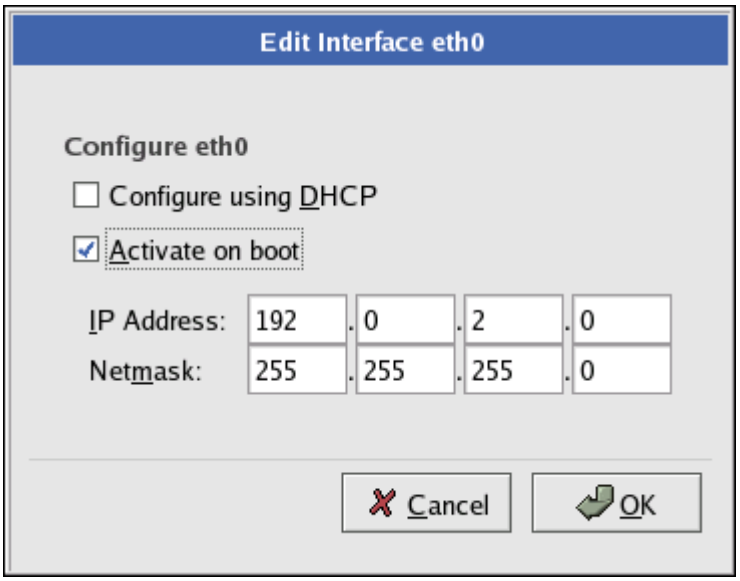
Choose upgrade an existing installation tab.

Select Skip boot loader updating option and click Next

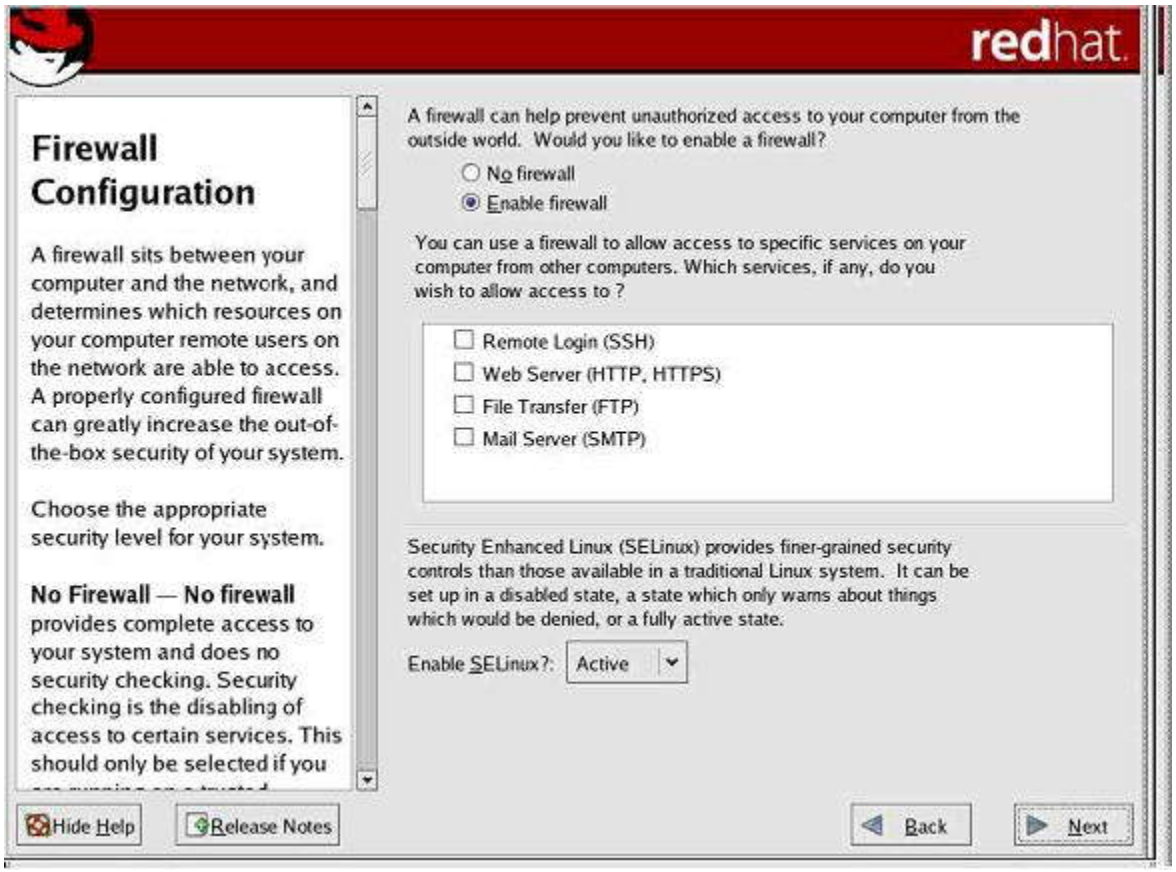
The installation program automatically detects any network devices you have and displays them in the **Network Devices** list.

Once you have selected a network device, Click **Edit** and configure the IP address and Netmask of the device via DHCP (or manually if DHCP is not selected) and you can choose to activate the device at boot time.





Enable and configure the firewall and SELinux and click Next. Use the default options unless necessary to allow access to the system



No firewall

No firewall provides complete access to your system and does no security checking.

Enable firewall

If you choose **Enable firewall**, connections are not accepted by your system (other than the default settings) that are not explicitly defined by you. By default, only connections in response to outbound requests, such as DNS replies or DHCP requests, are allowed.

Remote Login (SSH)

*S*ecure *Sh*ell (SSH) is a suite of tools for logging in to and executing commands on a remote machine.

Web Server (HTTP, HTTPS)

The HTTP and HTTPS protocols are used by Apache (and by other Web servers) to serve webpages.

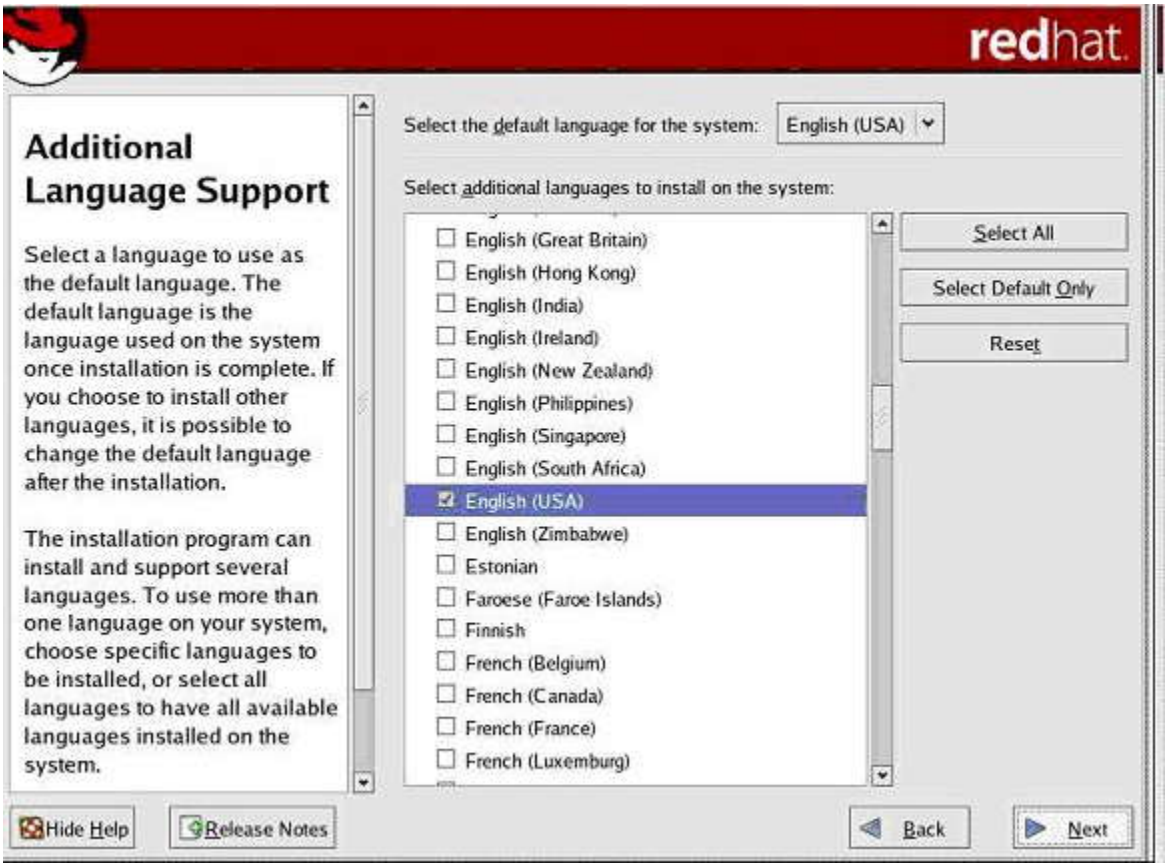
File Transfer (FTP)

The FTP protocol is used to transfer files between machines on a network. If you plan on making your FTP server publicly available, enable this option. You must install the vsftpd package in order to publicly serve files.

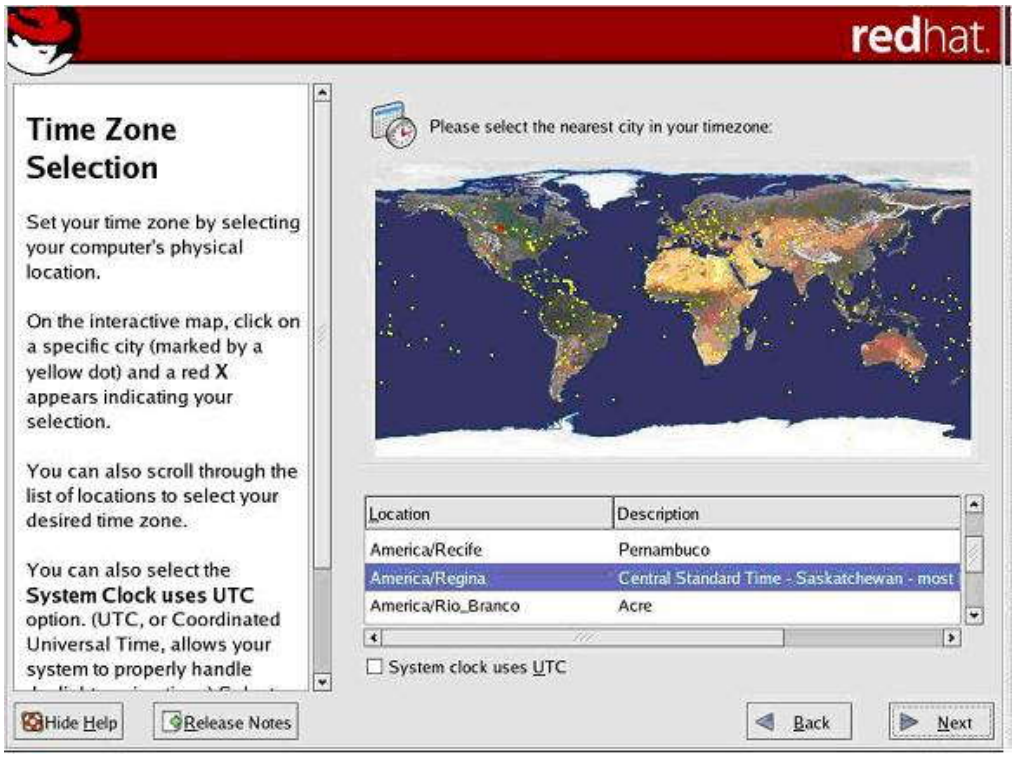
Mail Server (SMTP)

To allow incoming mail delivery through your firewall, so that remote hosts can connect directly to your machine to deliver mail.

Select any additional languages and click Next.



Select your time zone and click Next

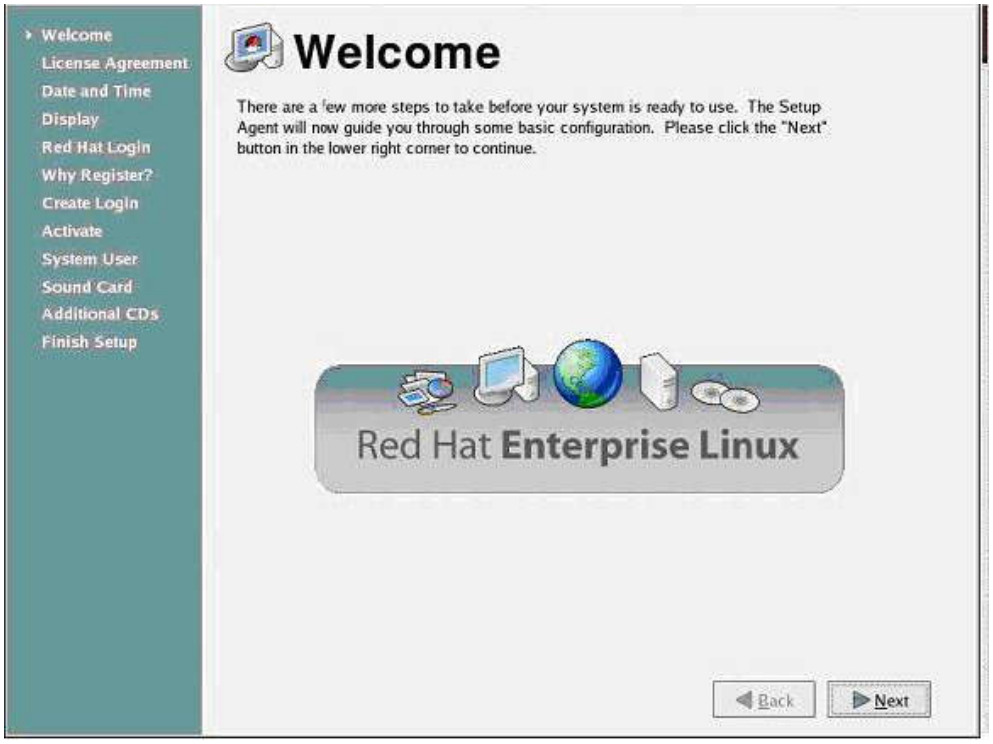


Set the root password and click Next.

Choose "Install the default packages," or if you know which packages you want to install "Customize software packages" and click Next.

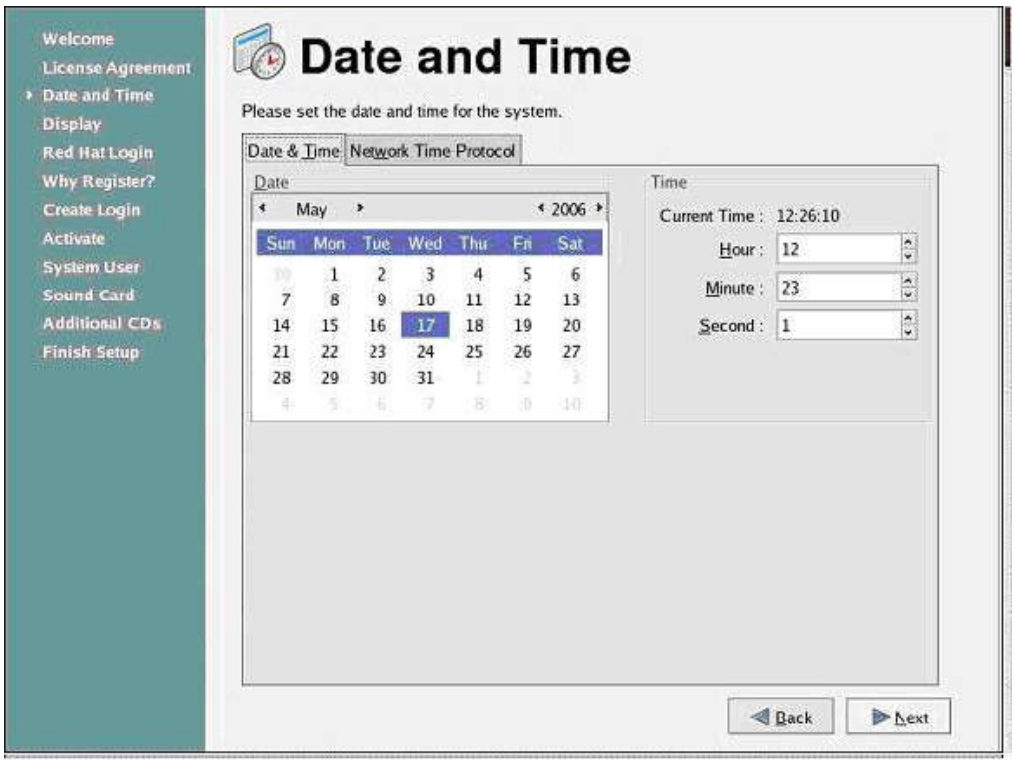
Click next to Install. Confirm that you have all of your CDs and click Continue.

After the installation is finished reboot the system and the Setup Agent will continue when you click Next.

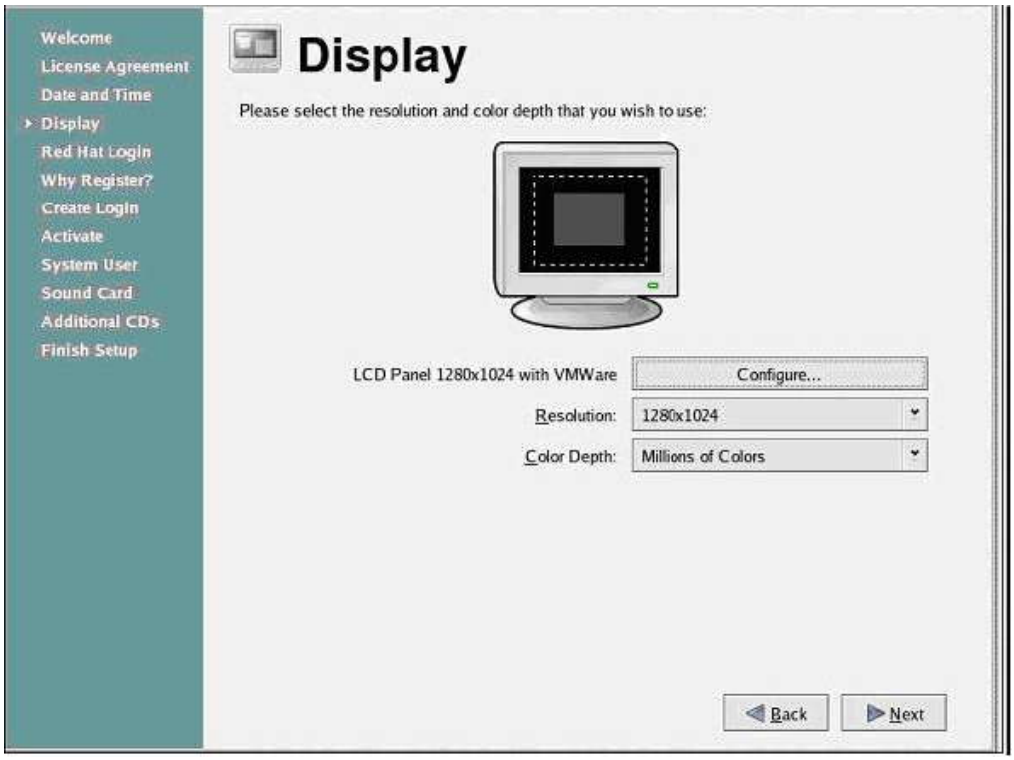


Read the license agreement and select "Yes, I agree to the License Agreement" and click Next.

Verify that the date and time are correct.



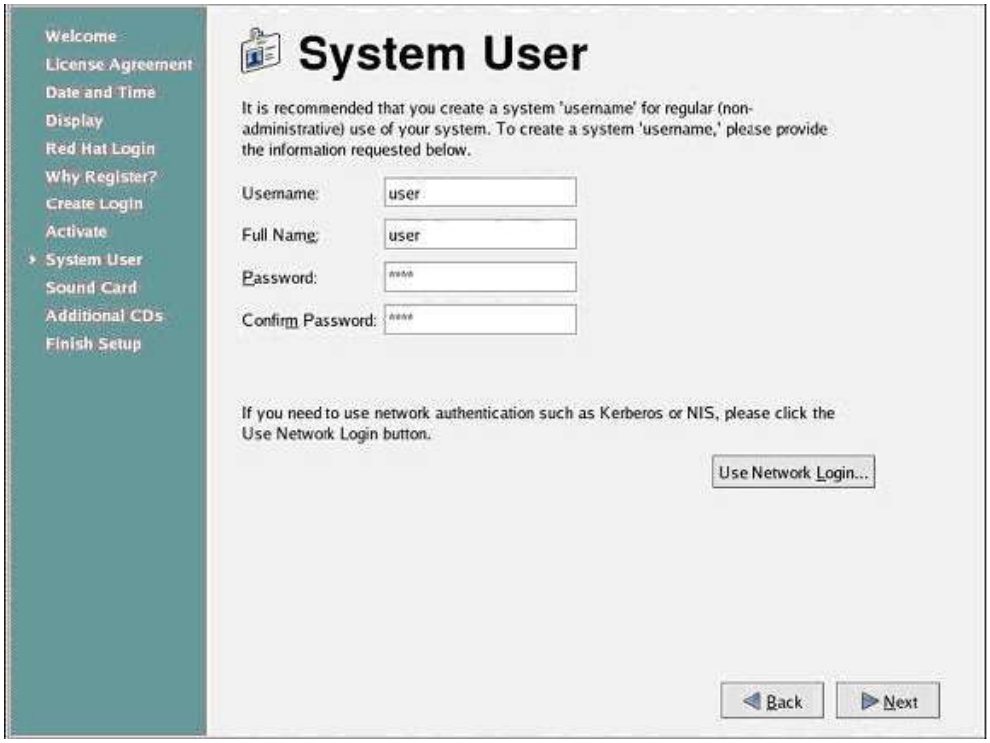
Configure your display properties



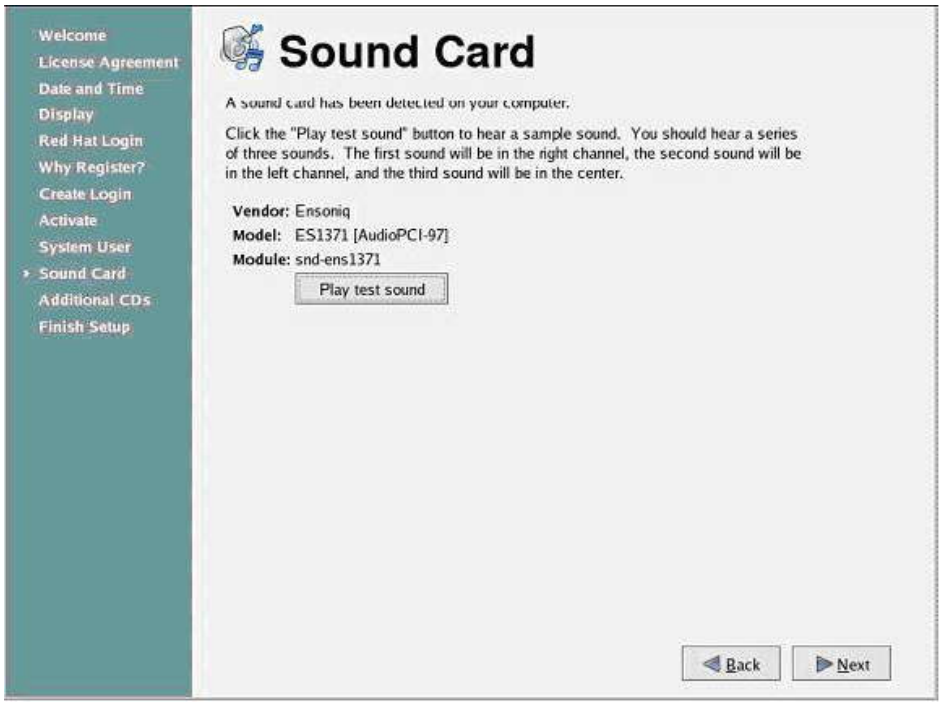
At the Red Hat Login screen choose "Tell me why I need to register and provide a Red Hat Login" and click Next.

Select "I can not complete the registration at this time. Remind me later." and click Next.

Enter your username and a password and click Next.



Test your sound card and click Next.

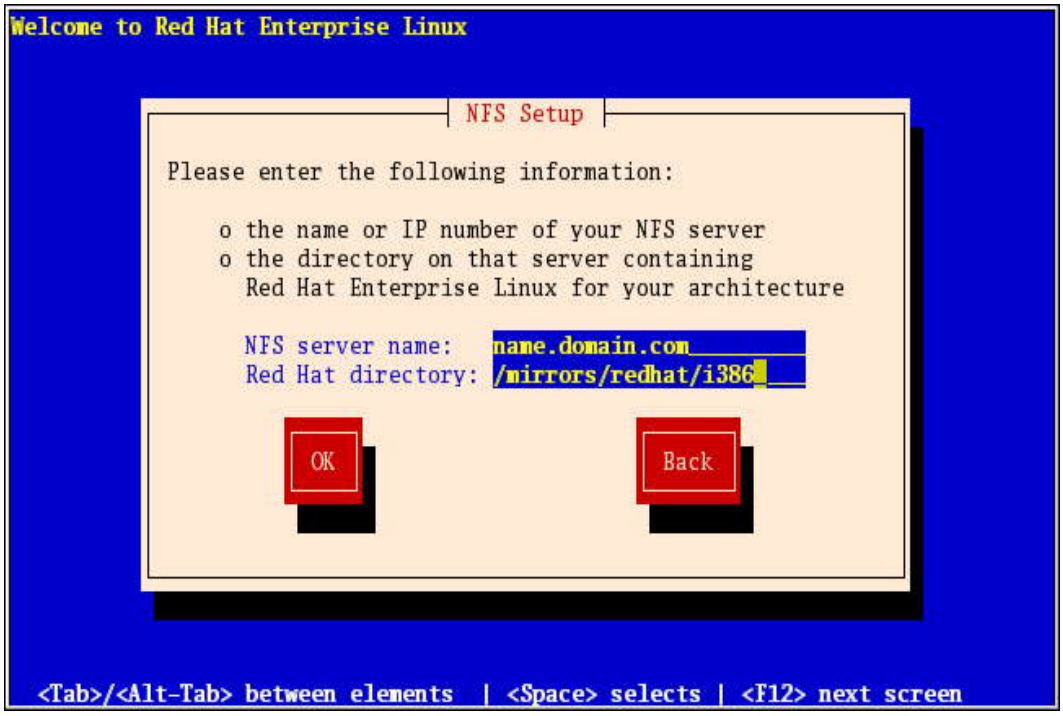


Install any additional software from CD or click Next.

Finish setup by clicking Next to reboot the system.

1.2 NFS Installation

(i) Boot the system from first CD. At the first screen press Linux askmethod and press enter.
(ii) Select the Language and Keyboard type
(iii) Select Installation method as NFS Image and configure the ipaddress and click ok to next screen.

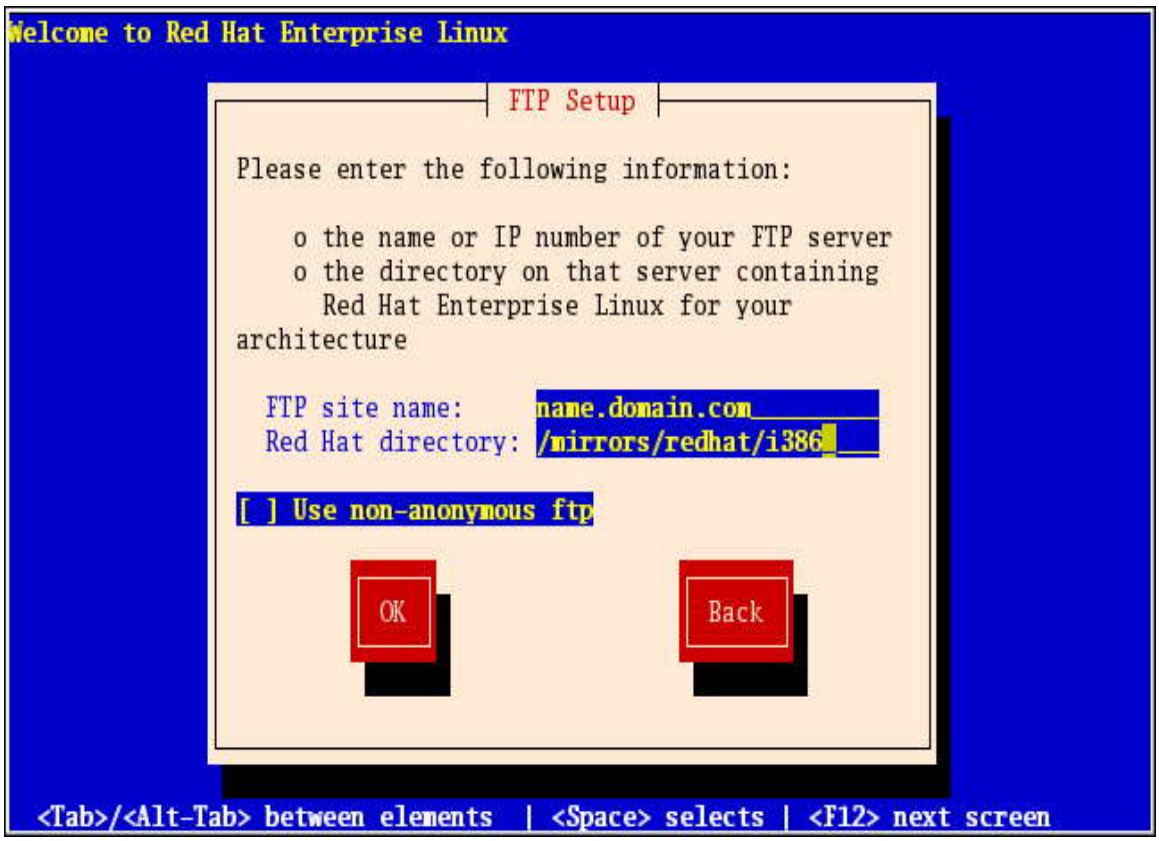


(iv) Enter the domain name or IP address of your NFS server. Next, enter the name of the exported directory.

(v) For example, if the NFS site contains the directory /mirrors/redhat/*arch*/RedHat/, enter /mirrors/redhat/*arch*/ where *arch* is replaced with the architecture type of your system, such as i386 and click ok to continue Installation.

1.3 FTP Installation

(i) Boot the system from first CD. At the first screen press Linux askmethod and press enter.
(ii) Select the Language and Keyboard type
(iii) Select Installation method as FTP and configure the ipaddress and click ok to next screen.

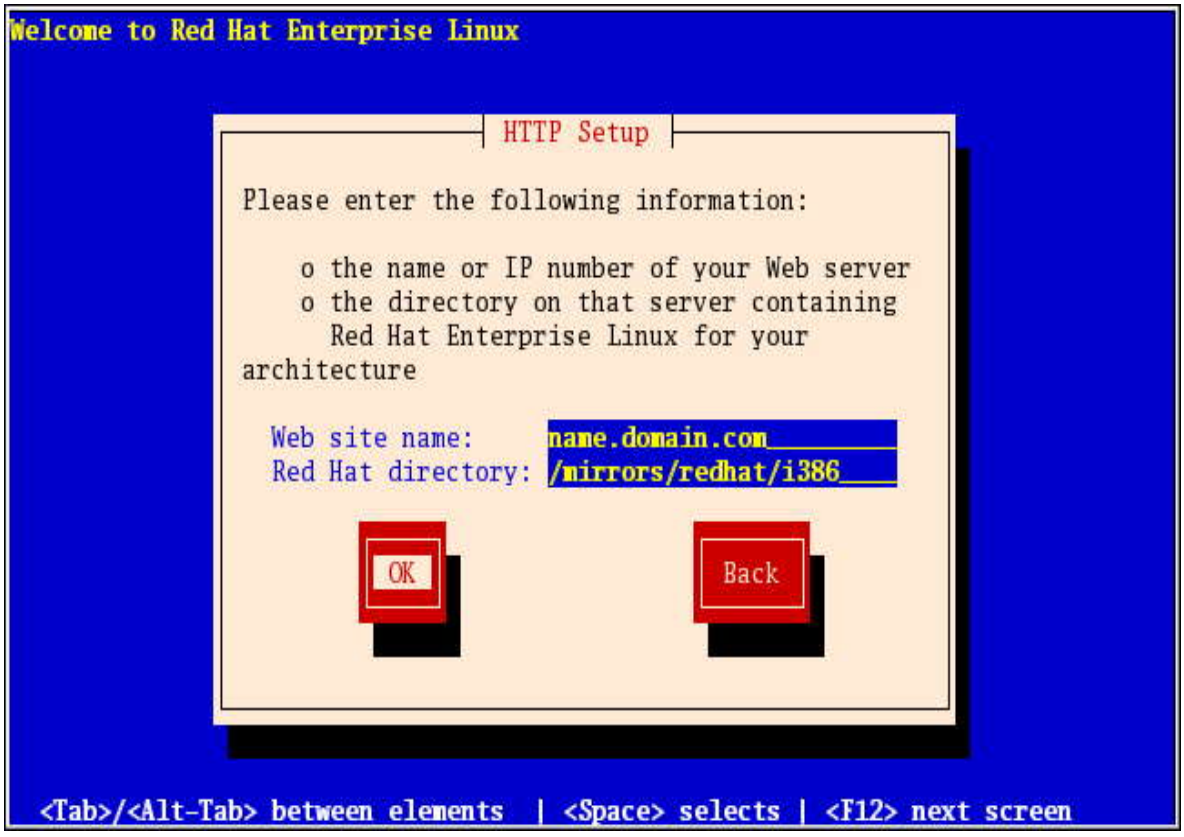


(iv) Enter the domain name or IP address of your FTP server. Next, enter the name directory containing the RedHat/ installation files
(v) For example, if the FTP site contains the directory /mirrors/redhat/*arch*/RedHat/, enter /mirrors/redhat/*arch*/ where *arch* is replaced with the architecture type of your system, such as i386 and click ok to continue installation.

1.4 HTTP Installation

(vi) Boot the system from first CD. At the first screen press Linux askmethod and press enter.

(vii) Select the Language and Keyboard type Select Installation method as HTTP and configure the ipaddress and click ok to next screen



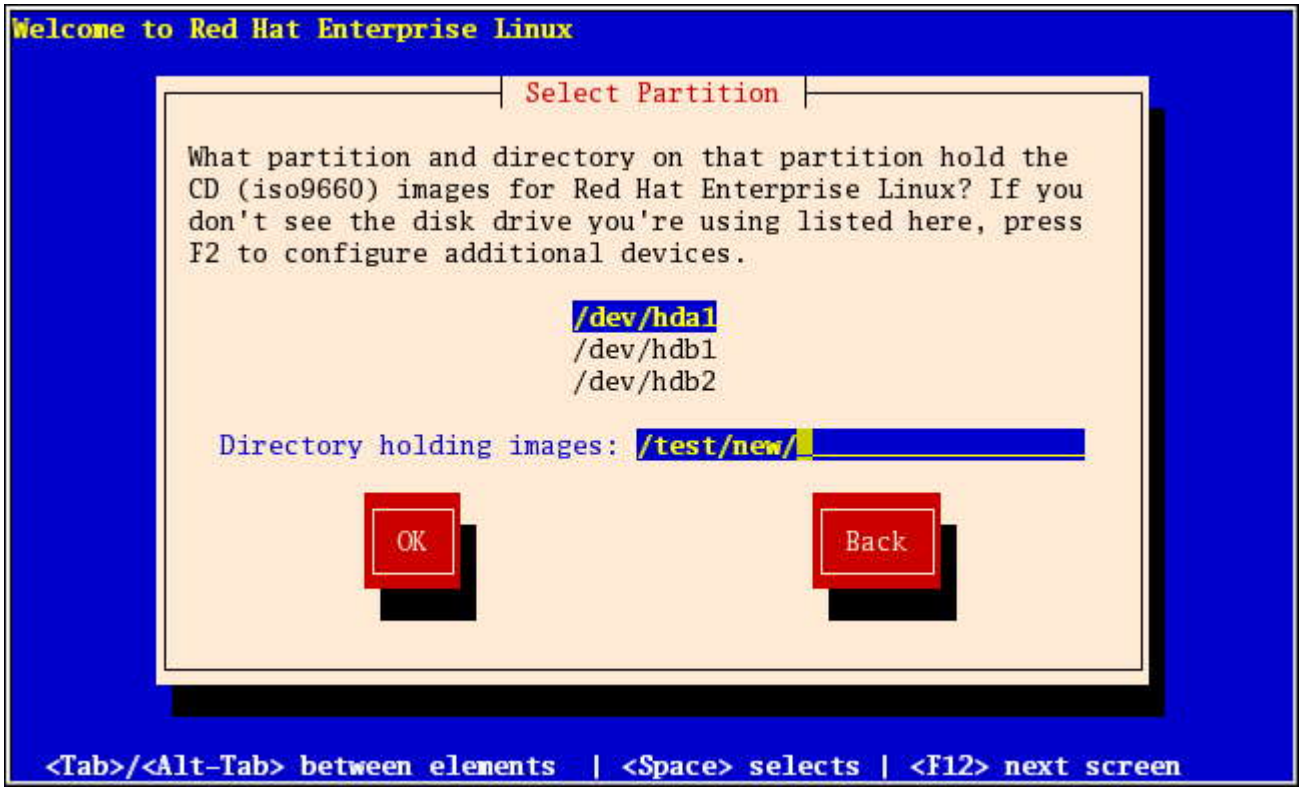
(viii) Enter the domain name or IP address of your HHTP server. Next, enter the name directory containing the RedHat/ installation files

(ix) For example, if the HTTP site contains the directory /mirrors/redhat/*arch*/RedHat/, enter /mirrors/redhat/*arch*/ where *arch* is replaced with the architecture type of your system, such as i386 and click ok to continue installation.

1.5 Hard Drive Installation

(x) Boot the system from first CD. At the first screen press Linux askmethod and press enter.

(xi) Select the Language and Keyboard type Select Installation method as Hard Drive Installation.



(xii) Enter the device name and partition containing the Redhat Enterprise Linux ISO images.

(xiii) For example, if the partition on which the ISO images is normally mounted as /home/ and the images are in /home/new you would enter /new/ and click ok to continue installation.

1.6 Kick Start Installation

(i) Kickstart installation can be performed using a local CD-ROM, a local hard drive, NFS, FTP, or HTTP.

(ii) Kickstart file is a simple text file, containing a list of items and it can be created using Kickstart Configurator (eg:system-config-kickstart)

(iii) If ks is used alone, the installation program will configure the Ethernet card in the system using DHCP. The system will use the "bootServer" from the DHCP response as an NFS server to read the kickstart file from (by default, this is the same as the DHCP server).

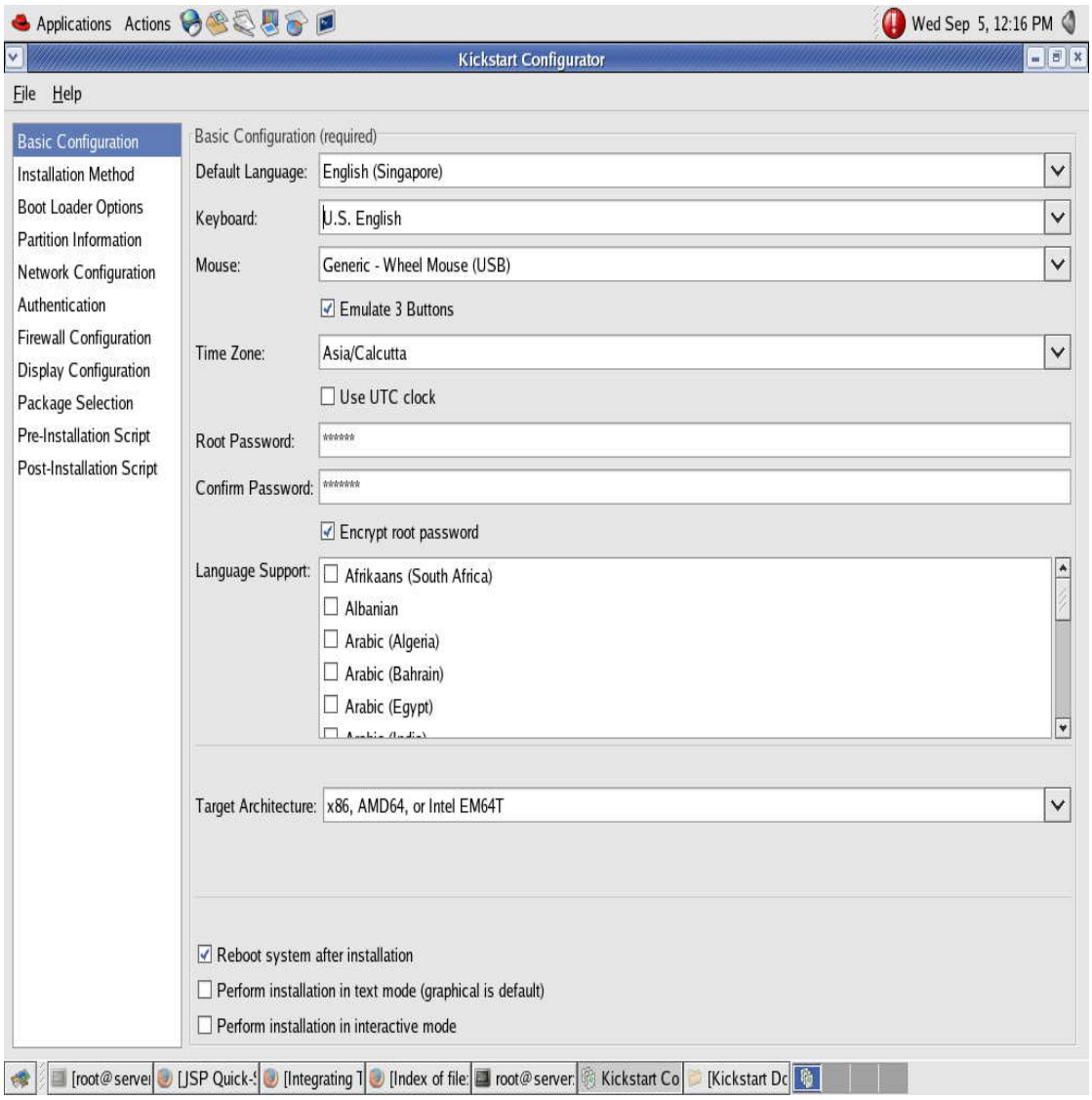
(iv) The name of the kickstart file is one of the following:

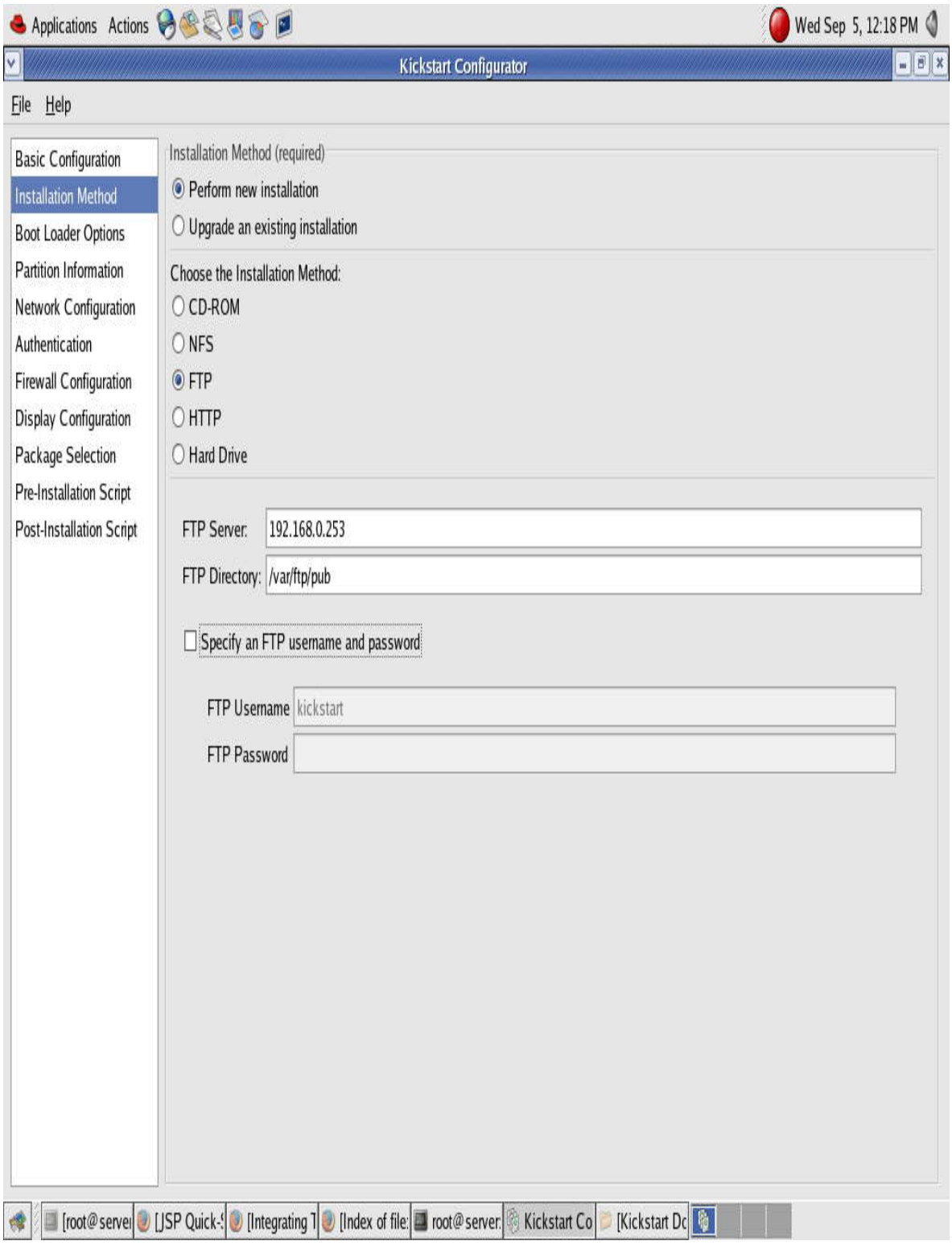
- If DHCP is specified and the bootfile begins with a /, the bootfile provided by DHCP is looked for on the NFS server.
- If DHCP is specified and the bootfile begins with something other then a /, the bootfile provided by DHCP is looked for in the /kickstart directory on the NFS server.
- If DHCP did not specify a bootfile, then the installation program tries to read the file /kickstart/1.2.3.4-kickstart, where *1.2.3.4* is the numeric IP address of the machine being installed.

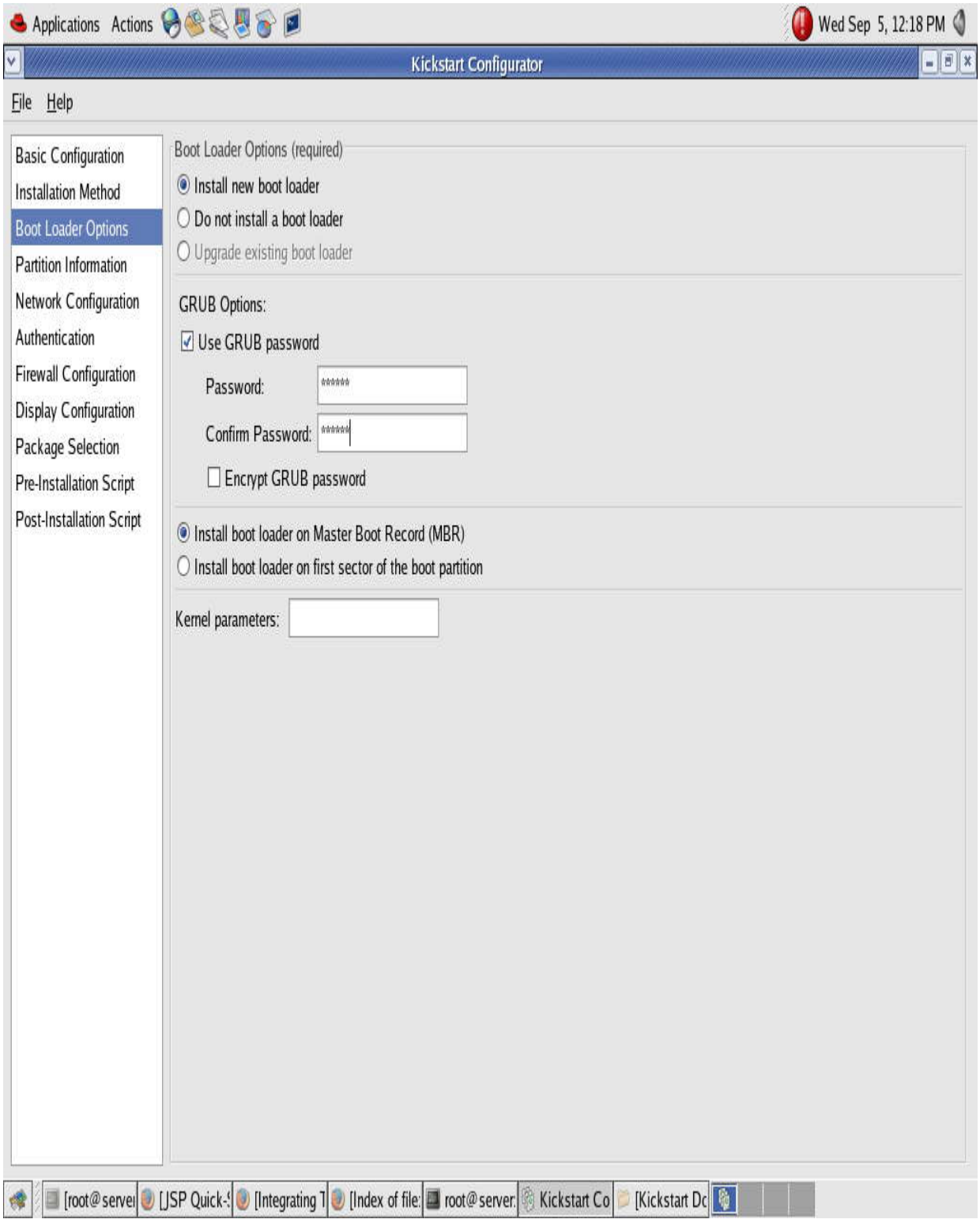
(iv) Copy the Installation tree of RedHat Linux CDROMs with the same directory structure.

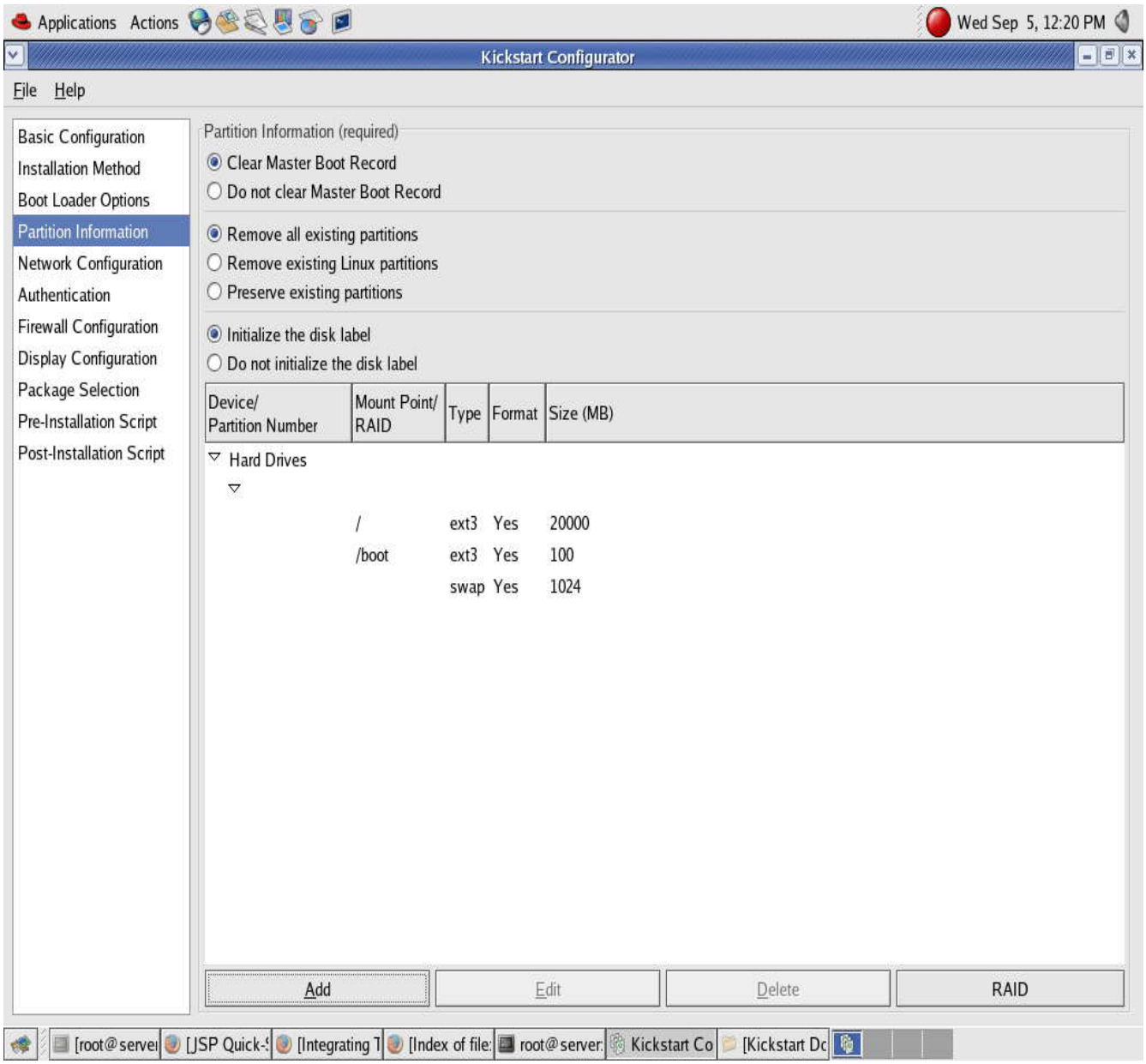
(iv) Now Boot the system from a Redhat Hat Linux CDROM 1 and enter
A special boot command at the boot prompt. The installation
Programs looks for a kickstart file if the ks command line arugument
Is passed to the kernel .
Example : Linux ks=nfs:<*server*>:/<*path*>

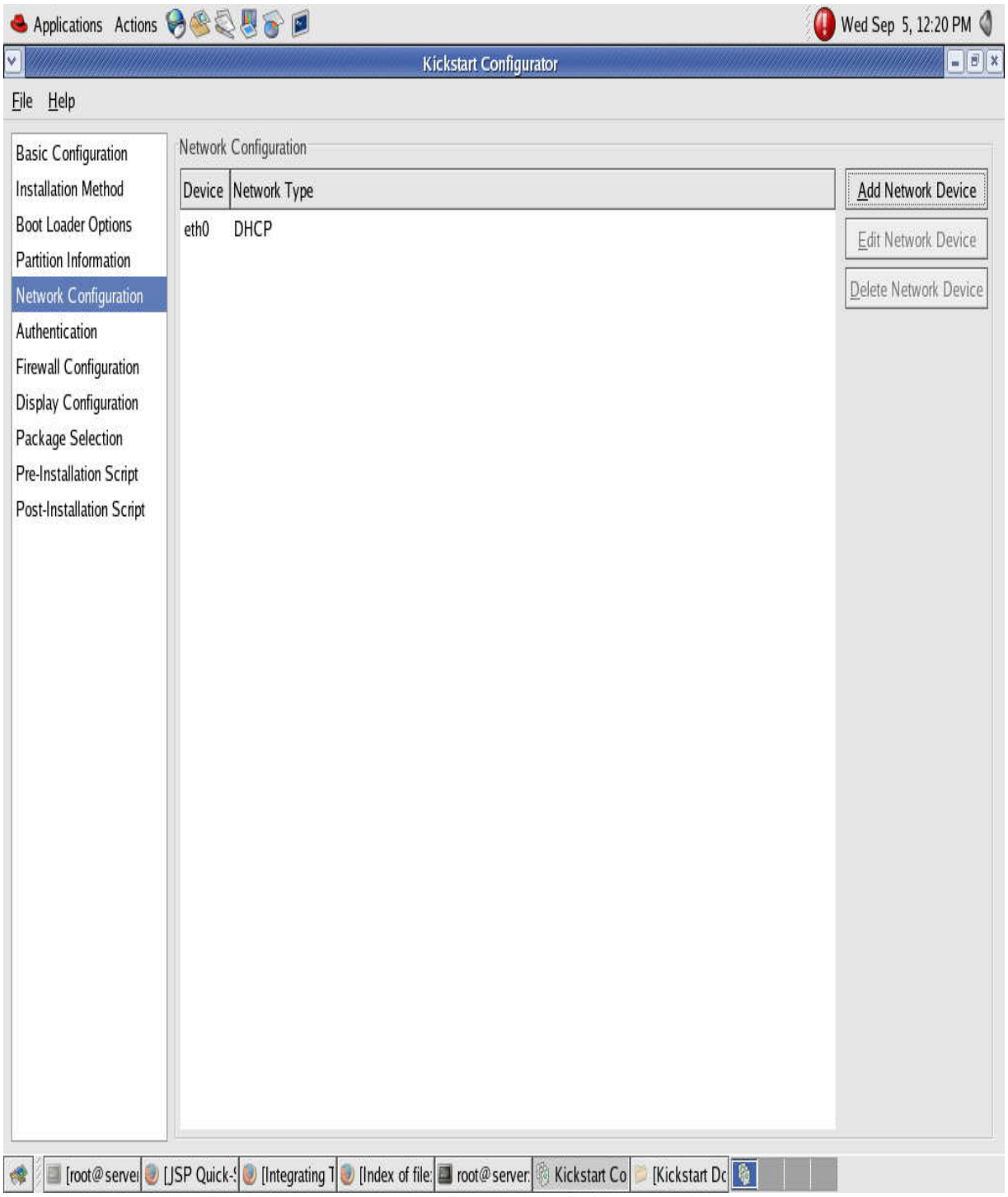
Creating kickstart file

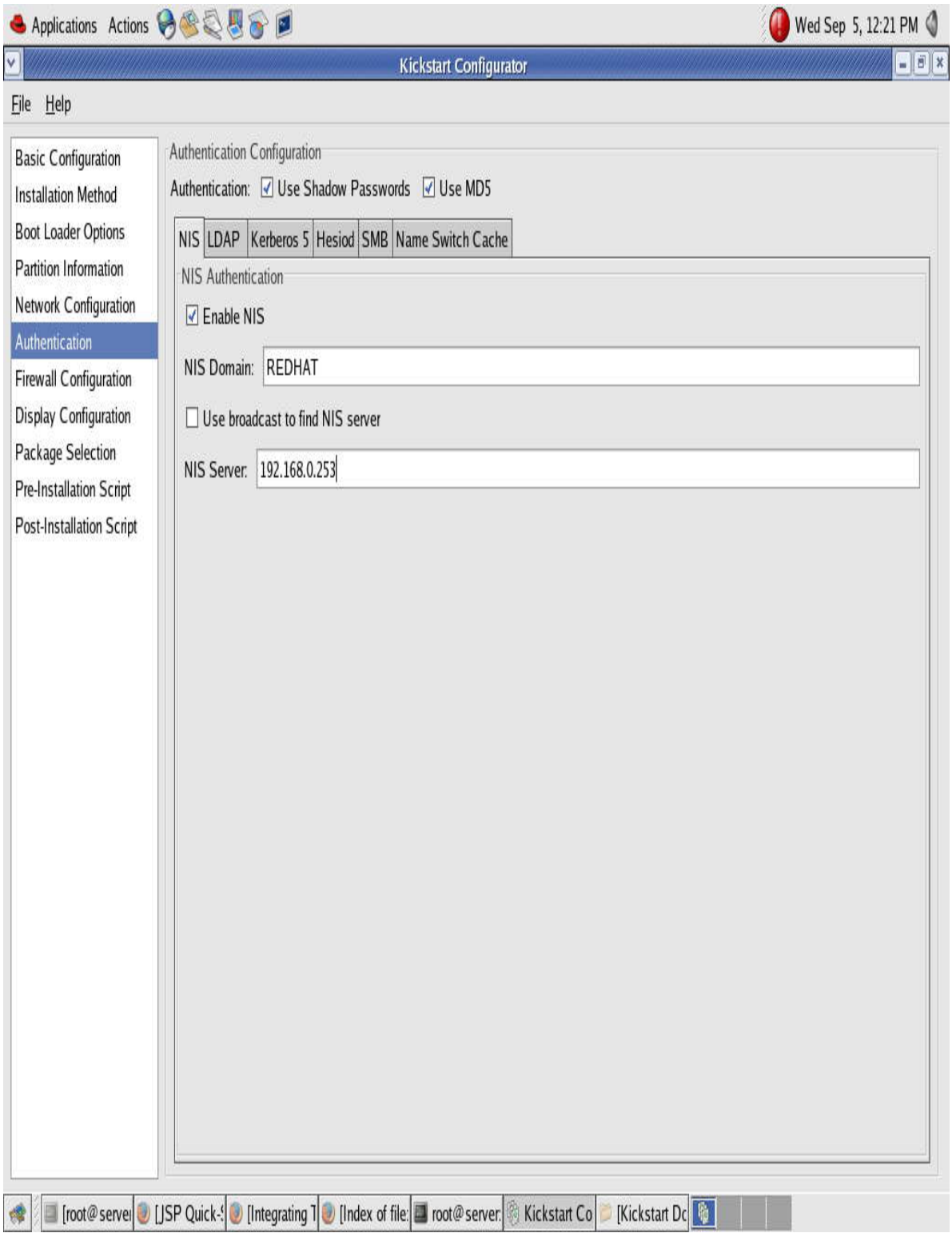

Applications Actions
Wed Sep 5, 12:16 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Basic Configuration (required)
Default Language: English (Singapore)
Keyboard: U.S. English
Mouse: Generic - Wheel Mouse (USB)
Emulate 3 Buttons
Time Zone: Asia/Calcutta
Use UTC clock
Root Password:
Confirm Password:
Encrypt root password
Language Support:
Afrikaans (South Africa)
Albanian
Arabic (Algeria)
Arabic (Bahrain)
Arabic (Egypt)
Target Architecture: x86, AMD64, or Intel EM64T
Reboot system after installation
Perform installation in text mode (graphical is default)
Perform installation in interactive mode

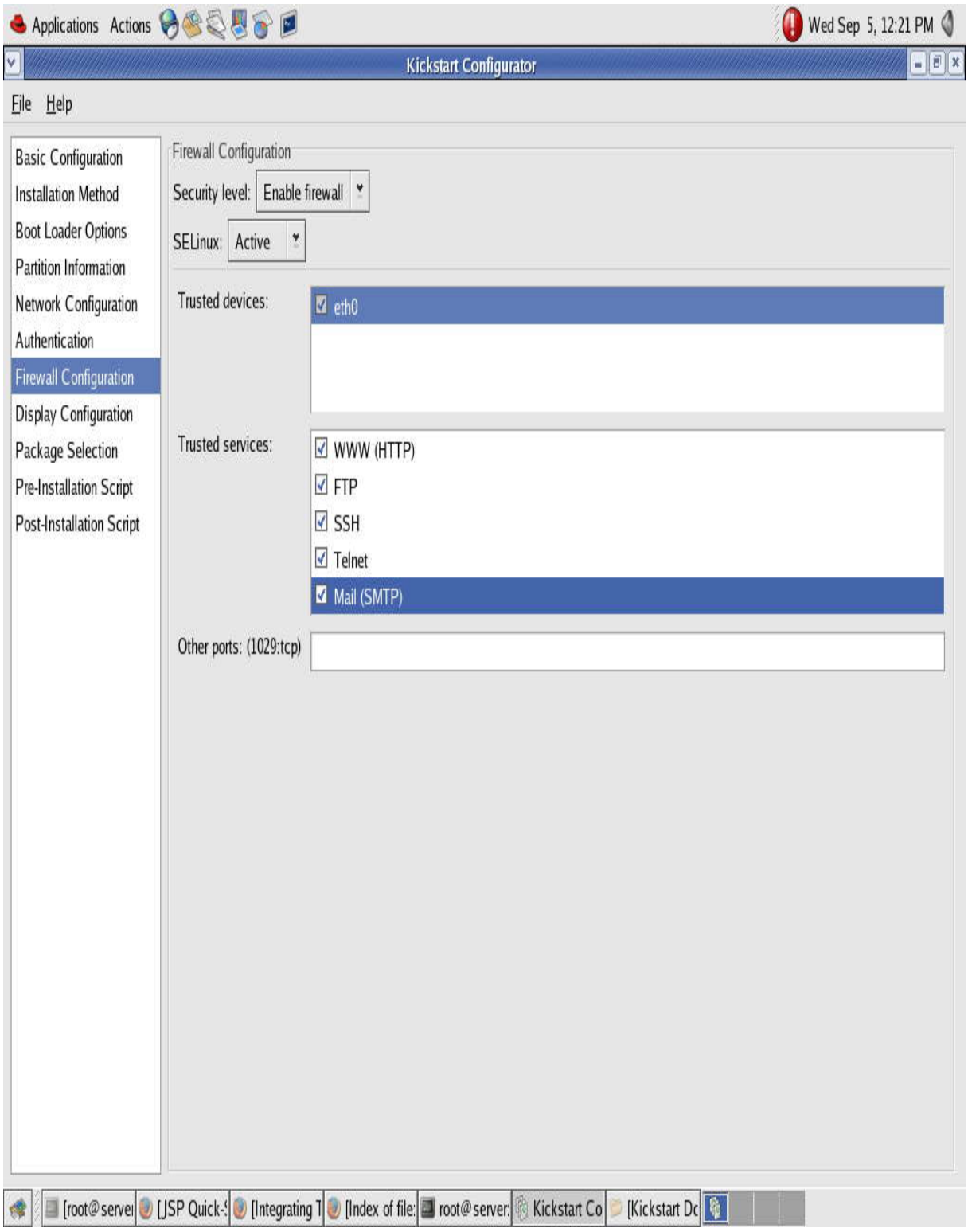

Applications Actions
Wed Sep 5, 12:18 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Installation Method (required)
Perform new installation
Upgrade an existing installation
Choose the Installation Method:
CD-ROM
NFS
FTP
HTTP
Hard Drive
FTP Server: 192.168.0.253
FTP Directory: /var/ftp/pub
Specify an FTP username and password
FTP Username kickstart
FTP Password
[root@server
[JSP Quick-
[Integrating T
[Index of file:
root@server:
Kickstart Co
[Kickstart Dc

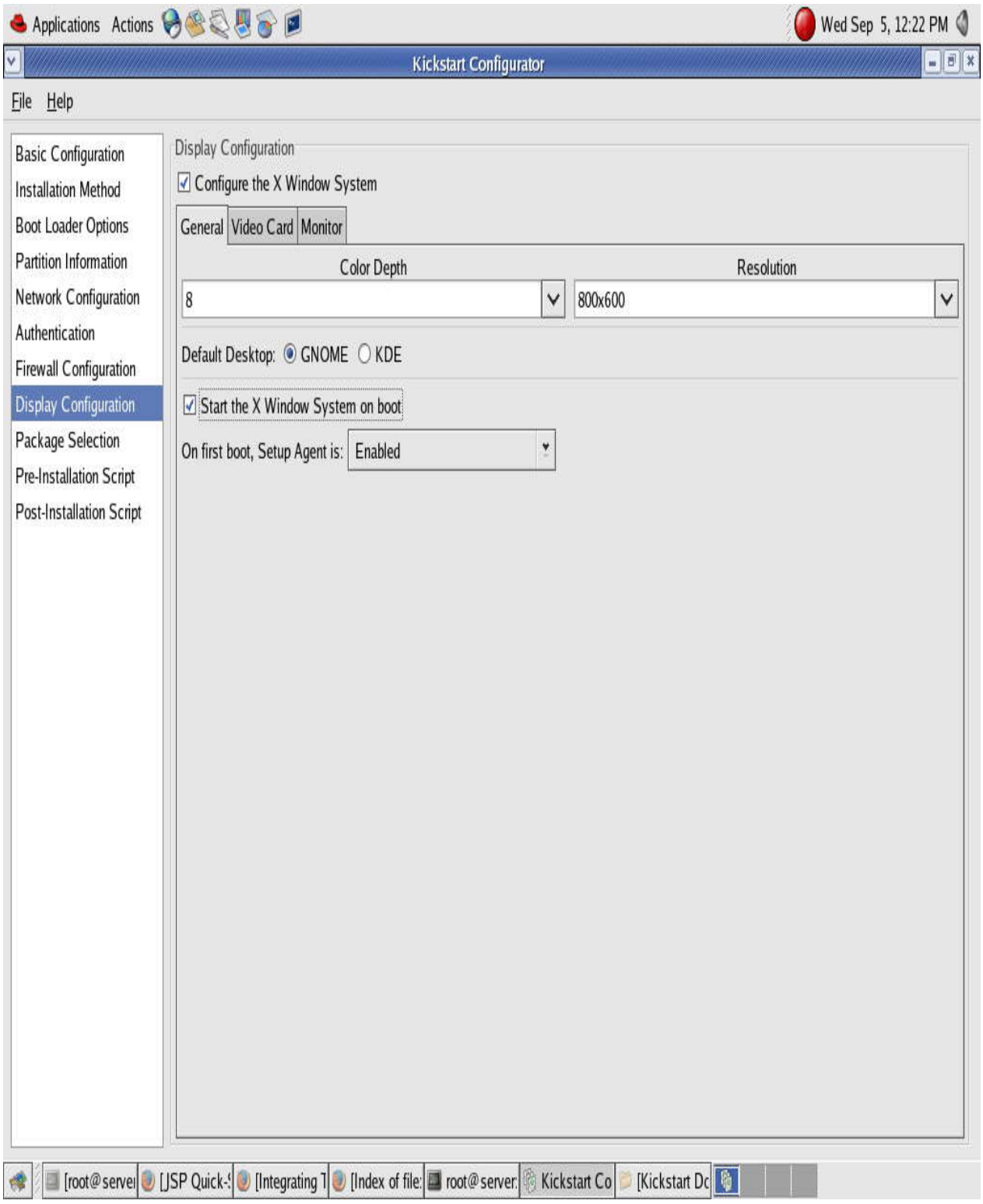

Applications Actions
Wed Sep 5, 12:18 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Boot Loader Options (required)
Install new boot loader
Do not install a boot loader
Upgrade existing boot loader
GRUB Options:
Use GRUB password
Password:
Confirm Password:
Encrypt GRUB password
Install boot loader on Master Boot Record (MBR)
Install boot loader on first sector of the boot partition
Kernel parameters:

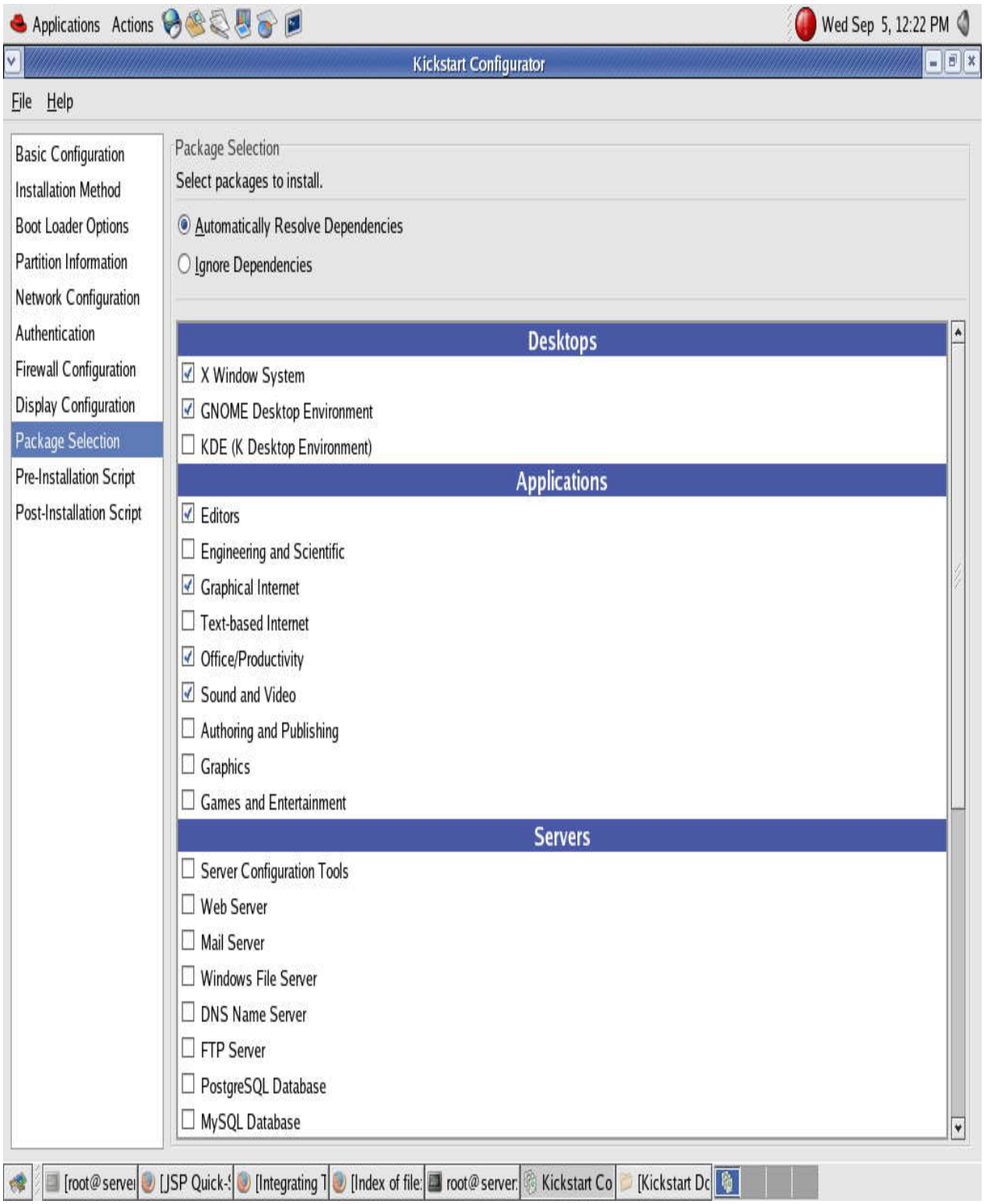

Applications Actions
Wed Sep 5, 12:20 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Partition Information (required)
Clear Master Boot Record
Do not clear Master Boot Record
Remove all existing partitions
Remove existing Linux partitions
Preserve existing partitions
Initialize the disk label
Do not initialize the disk label
Device/ Partition Number
Mount Point/ RAID
Type
Format
Size (MB)
Hard Drives
/ ext3 Yes 20000
/boot ext3 Yes 100
swap Yes 1024
Add
Edit
Delete
RAID

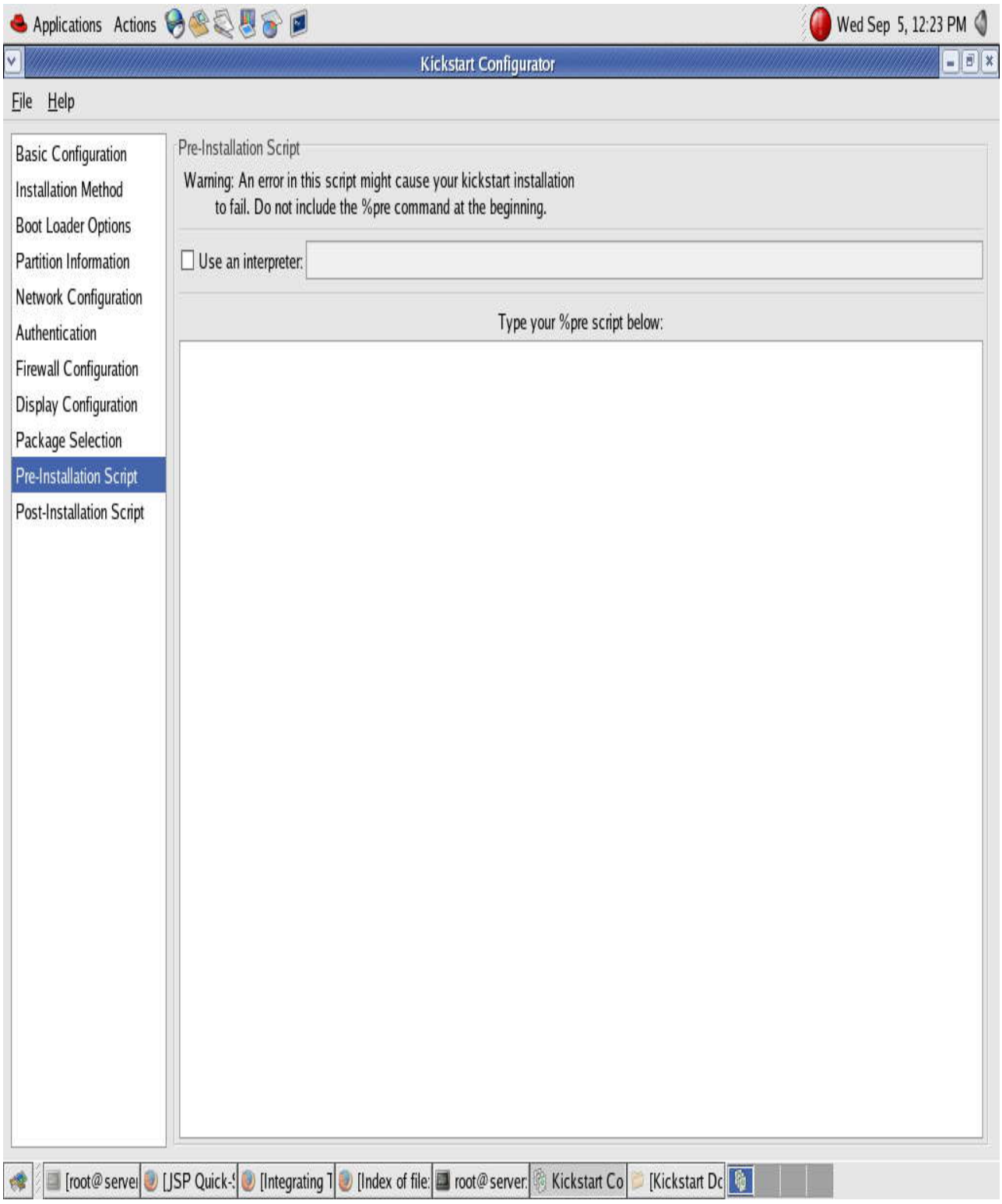

Applications Actions
Wed Sep 5, 12:20 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Network Configuration
Device
Network Type
eth0 DHCP
Add Network Device
Edit Network Device
Delete Network Device

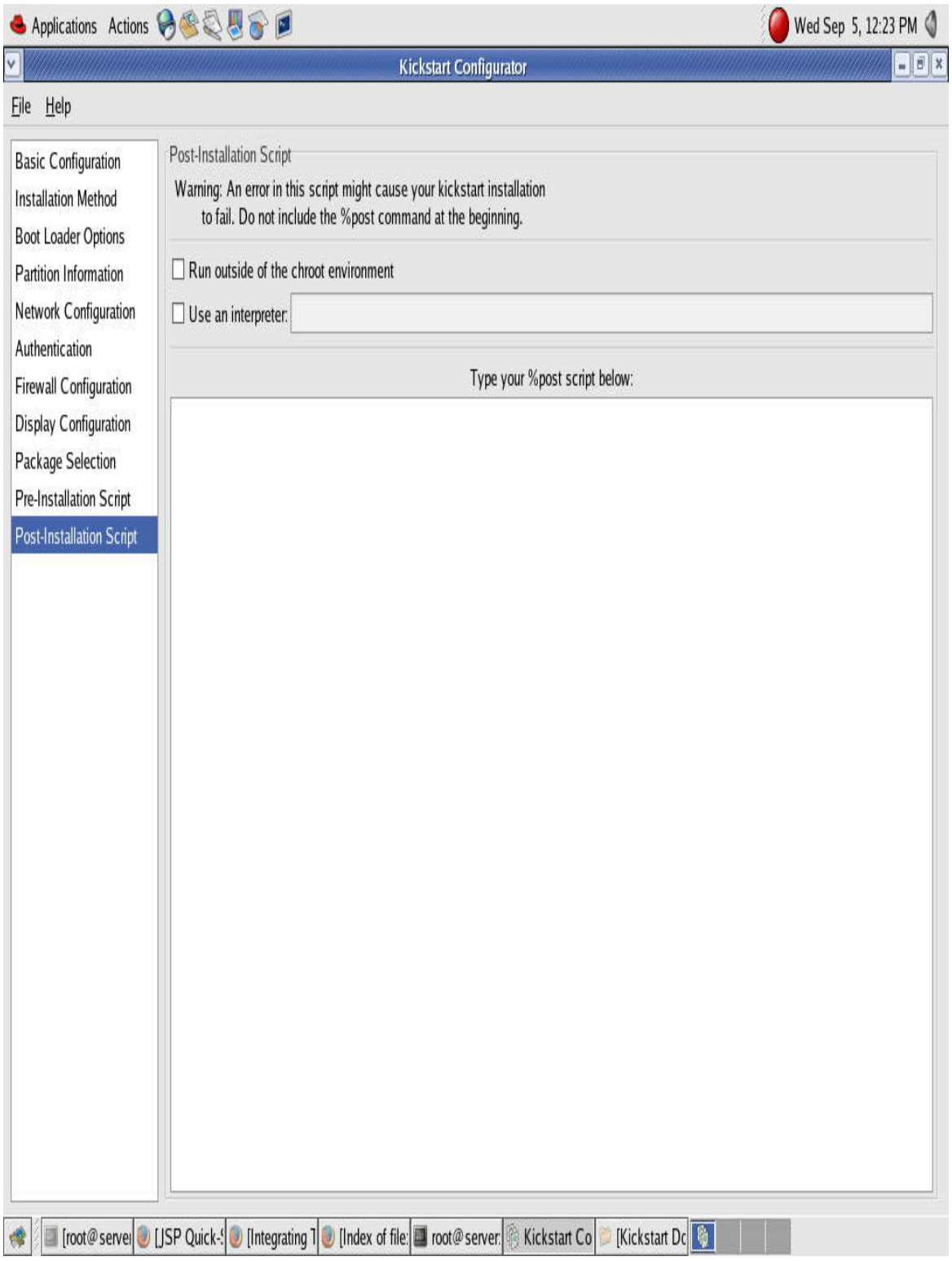

Applications Actions
Wed Sep 5, 12:21 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Authentication Configuration
Authentication: Use Shadow Passwords Use MD5
NIS LDAP Kerberos 5 Hesiod SMB Name Switch Cache
NIS Authentication
Enable NIS
NIS Domain: REDHAT
Use broadcast to find NIS server
NIS Server: 192.168.0.253


Applications Actions
Wed Sep 5, 12:21 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Firewall Configuration
Security level: Enable firewall
SELinux: Active
Trusted devices:
eth0
Trusted services:
WWW (HTTP)
FTP
SSH
Telnet
Mail (SMTP)
Other ports: (1029:tcp)
[root@server
[JSP Quick-S
[Integrating T
[Index of file:
root@server:
Kickstart Co
[Kickstart Dc


Applications Actions
Wed Sep 5, 12:22 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Display Configuration
Configure the X Window System
General
Video Card
Monitor
Color Depth
Resolution
8
800x600
Default Desktop: GNOME KDE
Start the X Window System on boot
On first boot, Setup Agent is: Enabled


Applications Actions
Wed Sep 5, 12:22 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Package Selection
Select packages to install.
Automatically Resolve Dependencies
Ignore Dependencies
Desktops
X Window System
GNOME Desktop Environment
KDE (K Desktop Environment)
Applications
Editors
Engineering and Scientific
Graphical Internet
Text-based Internet
Office/Productivity
Sound and Video
Authoring and Publishing
Graphics
Games and Entertainment
Servers
Server Configuration Tools
Web Server
Mail Server
Windows File Server
DNS Name Server
FTP Server
PostgreSQL Database
MySQL Database


Applications Actions
Wed Sep 5, 12:23 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Pre-Installation Script
Warning: An error in this script might cause your kickstart installation to fail. Do not include the %pre command at the beginning.
Use an interpreter:
Type your %pre script below:
[root@server
[JSP Quick-
[Integrating T
[Index of file:
root@server:
Kickstart Co
[Kickstart Dc


Applications Actions
Wed Sep 5, 12:23 PM
Kickstart Configurator
File Help
Basic Configuration
Installation Method
Boot Loader Options
Partition Information
Network Configuration
Authentication
Firewall Configuration
Display Configuration
Package Selection
Pre-Installation Script
Post-Installation Script
Post-Installation Script
Warning: An error in this script might cause your kickstart installation to fail. Do not include the %post command at the beginning.
Run outside of the chroot environment
Use an interpreter:
Type your %post script below:
[root@server
[JSP Quick-
[Integrating T
[Index of file:
root@server:
Kickstart Co
[Kickstart Dc

4. Startup and Shutdown

1. Linux Booting Sequence:

Booting sequence is nothing but the order of events happen while you start up Linux System till you receive login prompt.

Here is the sequence of events:

1. POST
2. BIOS
3. Master Boot Record (MBR)
4. LILO or GRUB
5. Kernel
6. init
7. Run Levels

1. POST: Power
   On Self Test is diagnostic software, residing at BIOS, testing all hardware required for proper booting of system.

2. BIOS:
   Basic Input/Output Devices is driver software for all basic input output devices. Once POST found that all minimum require hardware for booting up the system are working fine, then BIOS loads driver software for all basic IO devices like keyboard, monitor, floppy disk and hard disk.

3. MBR:
   After loading driver software for all basic IO devices, system reads track0, sector 1 of available boot device, and gets the location of operating system to boot. MBR normally points to the Boot Loader.

4. LILO or GRUB:
   These two are Linux boot loaders, which facilitates dual boot. These boot loaders allows selecting the operating system you would like to boot. This will also allows you to boot from different Linux kernels.

   Normally Boot loader performs the following:
   - Loads /boot/boot.b
   - prompt for (or timeout to default) partition or kernel

   Then control will be given to the partition or kernel that you have selected for booting

5. kernel:
   After loading second-stage boot loader (/boot/boot.b) and it finds the kernel image (as you have selected as mentioned above), kernel will be loaded (runs).

Then kernel performs the following tasks:

a. initialise devices and optionally loads initrd,
b. mounts root filesystem specified by bootloader
c. kernel prints: VFS: Mounted root (ext2 filesystem) readonly.
d. runs /sbin/init which is process number 1 (PID=1)
e. init prints: INIT: version x.xx booting

f. initrd
g. Allows setup to be performed before root FS is mounted
h. Bootloader loads ram disk image
i. kernel runs /linuxrc
j. load modules
k. initialise devices
l. /linuxrc exits
m. "real" root is mounted
n. kernel runs /sbin/init

6. init:

/sbin/init reads /etc/inittab for instructions on how to proceed. The first thing that init looks for in inittab is, a line with the keyword 'sysinit' in the action field. This is the command that should be executed when the system is booting. As the line looks as below:

System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
It run the script / etc/rc.d/rc.sysinit

This run /sbin/initlog, devfs to generate/manage system devices, network scripts: /etc/sysconfig/network, load keymap, system fonts, Mount /proc and start device controllers, Re-mount root file system as read/write, Direct kernel to load kernel parameters and modules: sysctl, depmod, modprobe, Set up clock: /etc/sysconfig/clock, Perform disk operations based on fsck configuration Check/mount/check/enable quotas non-root file systems: fsck, mount, quotacheck, quotaon, Initialize logical volume management: vgscan, /etc/lvmtab, Activate syslog, write to log files: dmesg, Configure sound: sndconfig, Activate PAM and Activate swapping: swapon.

If you would like to configure any Local system boot processes, this can be placed in /etc/rc.d/rc.local file. This file will be run after /etc/rc.d/rc.sysinit

sbin/init program then starts the system as defined by the initdefault directive in the /etc/inittab file.

id:5:initdefault:

In this example a runlevel of "5" is chosen. One of these process started by init is /sbin/rc. This script runs a series of scripts in the directories /etc/rc.d/rc0.d/, /etc/rc.d/rc1.d/, /etc/rc.d/rc2.d/, etc according to the run level in which system is going to start.

7. Runlevels

Run levels used by Linux system are as below:

	Scripts Directory
0 --- halt	/etc/rc.d/rc0.d/
1 --- Single user mode	/etc/rc.d/rc1.d/
2 --- Multi-user mode without NFS	/etc/rc.d/rc2.d/
3 --- Multi-user mode with all features	/etc/rc.d/rc3.d/
4 --- Unused	/etc/rc.d/rc4.d/
5 --- X11	/etc/rc.d/rc5.d/
6 --- Reboot	/etc/rc.d/rc6.d/

s or S --- Single user mode

Scripts in these directories are executed for each boot state of operation until it becomes fully operational. Scripts beginning with S denote startup scripts while scripts beginning with K denote shutdown (kill) scripts. Numbers follow these letters to denote the order of execution. (lowest to highest)

Shutdown Commands:

Shutdown:
- init 0
- shutdown -h now
  - -h halt after shutdown
  - -c: Cancel scheduled shutdown
- halt
- poweroff

Reboot:
- init 6
- shutdown -r now
- reboot

Enter single user mode:
- init 1

2. Different Boot Loader:

On a Red Hat Linux system, the boot loader's function is to locate the Linux kernel, and any other necessary files, and load them into memory. It then starts the kernel s o that the kernel can run processes. The boot loader also lets you control how a system is booted. If you dual boot, a boot loader enables you to choose between operating systems on startup.

There are different boot managers available to load Linux. The most common boot managers used by Intel users are Linux Loader (LILO), and the Grand Unified Bootloader (GRUB), BOOTLIN, SYSLINUX and more third party utilities available in the market. In order to have dual boot Linux with Solaris is SPARC machine you can use SILO. Alpha users can use MILO to have dual boot with Linux. Now we are going to see in detail about installing and configuring the boot managers LILO, GRUB and BOOTLIN.

Steps to install GRUB:

GRUB is the default boot loader in RedHat Linux 8 and above. Here are the steps to install GRUB boot loader.

Step1: Download the source "grub-0.5.96.1.tar.gz"
Step2: tar -xzvf grub-0.5.96.1.tar.gz
Step3: cd grub-0.5.96.1
Step4: ./configure
Step5: make
Step6: make install

This will install grub on your system. When you restart the system, GRUB will provide you with the boot menu, so that you can select the operating system to boot from.

You can alos go to grub command prompt and you can boot from different kernel other than the default one as below:

```
grub> root (hd0,1)
grub> kernel /vmlinuz_new root=/dev/sda2 ro vga=791
grub> boot
```

Now it will boot from the kernel named /vmlinuz_new.

Uninstalling GRUB:

You can uninstall GRUB just by overwriting other boot loaders such as

LILO (Linux Loader)
Dos Boot Record (DOS Boot loader)

If you uninstall GRUB Dos boot record, you should have Linux startup floppy to boot your system into Linux. If you overwrite GRUB with LILO, this will take care of booting Linux system.

Overwriting GRUB with LILO

Running the command /sbin/lilo will install LILO on your system. The configuration file for LILO is /etc/lilo.conf.

Overwriting GRUB with Dos boot record:
---

you can action this, by performing the following steps:

Bootup your XP install disk
Go to the recovory console by pressing 'R'.
Here type the command "fixmbr"

This will uninstall GRUB from the MBR and installs XP boot loader.

You can also uninstall GRUB by booting your system with DOS boot floppy and running the command below;

A:\> fdisk /MBR

This will uninstall GRUB and overwrite MBR with the Dos boot Record.

Running the following command from DOS boot floppy will also overwrite the boot loader
A:\> sys C:

NOTE: You should have Linux startup floppy to boot from Linux, if you have uninstalled GRUB by overwriting Dos Boot Record.

Creating GRUB Boot Floppy:

In case, 'GRUB' boot loader corrupted or you would kike to boot from Linux after uninstalling 'GRUB', you may need to have GRUB boot floppy.

How will you create GRUB boot floppy?

Here are the steps:
Insert a floppy disk on the drive.

```
fdformat /dev/fd0
mkfs -t msdos /dev/fd0
mount -t msdos /dev/fd0 /floppy
mkdir -p /floppy/boot/grub
cp /usr/local/share/grub/i386-pc/stage* /floppy/boot/grub
```

Now go to GRUB prompt by running the GRUB executable (/sbin/grub). Run the following commands in the GRUB prompt:

```
grub> root (fd0)
grub> setup (fd0)
grub> quit
```

Now floppy is ready for booting Linux and it will provide you with the boot menu. You can select the operating system to be booted, so that system control will be passed on to hard disk, and system will continue with the booting.

3. Kernel Modules

To insert, remove and list Linux Loadable Kernel Modules (LKM):

The basic commands for inserting and removing LKM are insmod & rmmod. To Insert Linux LKM serial.o, type the following command:

```
#insmod serial.o
```

serial.o contains the device driver for serial port. The command to remove a LKM from the kernel rmmod

```
#rmmod serial
```

This will remove the specified module from the kernel. The command to list all loaded LKMs is lsmod

```
#lsmod
```

You can list all the presently loaded Linux kernel modules by viewing /ptoc/nodules file. This file contains list of all loaded Linux loadable kernel modules.

Intelligent Loading Of Linux Loadable Kernel Modules- Modprobe:

You can perform loading and unloading of Linux Loadable kernel Modules using the higher level program modprobe. The main thing that modprobe does is automatically load the prerequisites of an LKM you request. It does this with the help of a file that you create with depmod and keep on your system.

```
#modprobe msdos
```

This performs an insmod of msdos.o, but before that it loads all prerequisites needed for loading msdos.o module.

The other major thing modprobe does for you is to find the object module containing the LKM given just the name of the LKM. For example, modprobe eth0 loads the appropriate network device driver to create and drive your eth0 device, assuming you set that up properly in modules.conf. The configuration file for modprobe is /etc/modules.conf. modprobe is especially important because it is the default program that the kernel module loader uses to load an LKM on demand. So if you use automatic module loading, you will need to set up modules.conf properly, otherwise things will not work.

4. Init process / Run Levels / Run control scripts

Init Process:

During booting, kernel loads /sbin/init, which first reads /etc/inittab for instructions on how to proceed. The first thing that init looks for in inittab is, a line with the keyword 'sysinit' in the action field. This is the command that should be executed when the system is booting. As the line looks as below:

```
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
```

It run the script / etc/rc.d/rc.sysinit

This run /sbin/initlog, devfs to generate/manage system devices, network scripts: /etc/sysconfig/network, load keymap, system fonts, Mount /proc and start device controllers, Re-mount root file system as read/write, Direct kernel to load kernel parameters and modules: sysctl, depmod, modprobe, Set up clock: /etc/sysconfig/clock, Perform disk operations based on fsck configuration Check/mount/check/enable quotas non-root file systems: fsck, mount, quotacheck, quotaon, Initialize logical volume management: vgscan, /etc/lvmtab, Activate syslog, write to log files: dmesg, Configure sound: sndconfig, Activate PAM and Activate swapping: swapon.

If you would like to configure any Local system boot processes, this can be placed in /etc/rc.d/rc.local file. This file will be run after /etc/rc.d/rc.sysinit

sbin/init program then starts the system as defined by the initdefault directive in the /etc/inittab file.

```
id:5:initdefault:
```

In this example a runlevel of "5" is chosen. One of these process started by init is /sbin/rc. This script runs a series of scripts in the directories /etc/rc.d/rc0.d/, /etc/rc.d/rc1.d/, /etc/rc.d/rc2.d/, etc according to the run level in which system is going to start.

Runlevels:

Run levels used by Linux system are as below:

```
0 --- halt
1 --- Single user mode
2 --- Multi-user mode without NFS
3 --- Multi-user mode with all features
4 --- Unused
5 --- X11
6 --- Reboot
s or S   --- Single user mode
```

Run control scripts:

Run control scripts are scripts used to start or stop a service. Linux system maintains run control scripts under the following path:

/etc/init.d

The soft link is created for required scripts to each run level as below:

Run control scripts for init 0 → /etc/rc.d/rc0.d
Run control scripts for init 1 → /etc/rc.d/rc1.d
Run control scripts for init 2 → /etc/rc.d/rc2.d
Run control scripts for init 3 → /etc/rc.d/rc3.d
Run control scripts for init 4 → /etc/rc.d/rc4.d
Run control scripts for init 5 → /etc/rc.d/rc5.d
Run control scripts for init 6 → /etc/rc.d/rc6.d

Therefore, when Linux system enters into any run level, it will run the appropriate scripts for starting or stopping the appropriate service.

How to Use a Run Control Script to Stop or Start a Service

- Become superuser.
- Stop the system service.
  # /etc/init.d/filename stop
- Restart the system service.
  # /etc/init.d/filename start
- Verify that the service has been stopped or started.
  # pgrep -f service

How to Add a Run Control Script

If you want to add a run control script to start and stop a service, copy the script into the /etc/init.d directory. Then, create links in the rcn.d directory where you want the service to start and stop.

- Become superuser.
- Add the script to the /etc/init.d directory.

  # cp filename /etc/init.d
  # chmod 0744 /etc/init.d/filename
  # chown root:sys /etc/init.d/filename

- Create links to the appropriate rcn.d directory.

```
# cd /etc/init.d
# ln filename /etc/rc2.d/Snnfilename
# ln filename /etc/rcn.d/Knnfilename
```

- Verify that the script has links in the specified directories.

```
# ls /etc/init.d/ /etc/rc2.d/ /etc/rcn.d/
```

The following example shows how to add a run control script for the xyz service.

```
# cp xyz /etc/init.d
# chmod 0744 /etc/init.d/xyz
# chown root:sys /etc/init.d/xyz
# cd /etc/init.d
# ln xyz /etc/rc2.d/S100xyz
# ln xyz /etc/rc0.d/K100xyz
# ls /etc/init.d /etc/rc2.d /etc/rc0.d
```

5. Creating Istallation Boot disk

The steps to configure a USB pen drive to boot and install Red Hat Enterprise Linux perform the following steps:

1. Format the USB flash drive as one FAT partition.

```
#mkdosfs /dev/
```

2. Copy the contents of `/RedHat/isolinux/` from the first installation CD to the USB flash drive. **NOTE:** isolinux.bin, boot.cat and TRANS.TBL can be removed or deleted.
3. Rename `isolinux.cfg` to `syslinux.cfg`.
4. Copy `/RedHat/images/pxeboot/initrd.img` from the first installation CD to the USB flash drive.
5. **OPTIONAL:** To configure any boot settings, edit the `syslinux.cfg` on the USB flash drive. For example to configure the installation to use a kickstart file shared over NFS,specify the following:

```
#linux ks=nfs::://ks.cfg
```

6. Make the USB flash drive bootable. Be sure to UNMOUNT the USB flash device.

```
#umount /dev/
#syslinux /dev/
```

7. Install GRUB on the USB flash drive.

```
#mount /dev/ /path/to/local/USB/mount
#grub-install --root-directory=/path/to/local/USB/mount /dev/
```

8. Verify that the USB flash drive has a `/boot/grub` directory. If it does not, create the directory manually.

```
#cd /path/to/USB/local/mount
#mkdir -p /boot/grub
```

9. Create the `grub.conf` file. Below is a sample `grub.conf`:

```
default=0
timeout=5
root (hd1,0)
title
kernel /vmlinuz
initrd /initrd.img
```

10. Copy or confirm the created `grub.conf` file is on the `/boot/grub/` directory of the USB flash drive.

6. Creating Rescue Disk

The command **mkbootdisk** creates a boot floppy appropriate for the running system. This boot disk can be used to boot your Linux system when the boot loader is corrupted. The created boot disk looks for the root file system on the device suggested by /etc/fstab.

The command syntax is:

```
# mkbootdisk --device /dev/fd0 <kernel0release>
```

Find the kernel release by running the command 'uname -r'

```
# uname –r
2.6.9-34.ELsmp
#
# mkbootdisk --device /dev/fd0 2.6.9-34.ELsmp
```

This will create a Linux Boot floppy. At any time the system can boot using this floppy.

7. Booting into different runlevels and rescue mode

Changing Runlevels at Boot Time

- Under Red Hat Enterprise Linux, it is possible to change the default runlevel at boot time.
- If using LILO, access the boot: prompt by typing [Ctrl]-[X]. Then type:
  linux *<runlevel-number>*
- In this command, replace *<runlevel-number>* with either the number of the runlevel to boot into (**1** through **5**), or the words **single** or **emergency**.
- If using GRUB, the steps are :
  - In the graphical GRUB boot loader screen, select the **Red Hat Enterprise Linux** boot label and press [e] to edit it.
  - Arrow down to the kernel line and press [e] to edit it.
  - At the prompt, type the number of the runlevel to boot into (**1** through **5**), or the words **single** or **emergency** and press [Enter].
  - The GRUB screen reappears with the kernel information. Press the [b] key to boot the system.

Changing Runlevels at Linux prompt:

The command "init" can be used to change the run level at user prompt. The syntax is as below:

```
#init <run-level-number>
```

To view in which run level you are working, you can use the command "who –r". This command will list the current as well as the previous run level as below:

```
# who -r
run-level 5  2008-05-05 15:11              last=S
#
```

As the current run level is 5, and if you would like to change the run level to 3, you should run the following command:

```
# init 3
```

This will bring the system to run level 3.

The following command will restart into runlevel 3:

```
#shutdown –r3
```

Booting the system in Rescue mode:

- To boot into rescue mode, boot the system using one of the following methods:
  - Booting the system using installation boot CD-ROM.
  - Booting the system from other boot media, such as USB flash devices.

- Booting the system from the Red Hat Enterprise Linux CD-ROM #1.

- At the boot prompt, add the keyword **rescue** as a kernel parameter. For example, type the following command at the installation boot prompt:

  boot: **linux rescue**

After booting off a boot disk or Red Hat Linux CD-ROM #1 and providing a valid rescue image, you will see the following message:

```
The rescue environment will now attempt to find your Red Hat
Linux installation and mount it under the directory
/mnt/sysimage.  You can then make any changes required to your
system.  If you want to proceed with this step choose
'Continue'. You can also choose to mount your filesystem
read-only instead of read-write by choosing 'Read-only'.
If for some reason this process fails you can choose 'Skip'
and this step will be skipped and you will go directly to a
command shell.
```

If you select **Continue**, it will attempt to mount your file system under the directory /mnt/sysimage. If it fails to mount a partition, it will notify you. If you select **Read-Only**, it will attempt to mount your file system under the directory /mnt/sysimage, but in read-only mode. If you select **Skip**, your filesystem will not be mounted. Choose **Skip** if you think your filesystem is corrupted.

Once you have your system in rescue mode, a prompt appears on VC (virtual console) 1 and VC 2 (use the [Ctrl]-[Alt]-[F1] key combination to access VC 1 and [Ctrl]-[Alt]-[F2] to access VC 2):

sh-2.05a#

If you selected **Continue** to mount your partitions automatically and they were mounted successfully, you are in single-user mode. To mount a Linux partition manually inside rescue mode, create a directory such as */foo*, and type the following command:

mount -t ext3 */dev/hda5 /a*

In the above command, */a* is a directory that you have created and */dev/hda5* is the partition you want to mount. If the partition is of type ext2, replace ext3 with ext2. If you do not know the names of your partitions, use the following command to list them:

fdisk –l

If your filesystem is mounted and you want to make your system the root partition, use the command chroot /mnt/sysimage. This is useful if you need to run commands such as rpm that require your root partition to be mounted as /. To exit the chroot environment, type exit, and you will return to the prompt.

8. understanding /etc/inittab file

- This file **etc/inittab** describes how the INIT process should set up the system in a certain run-level.
- Runlevels are a state, or *mode*, defined by the services listed in the SysV `/etc/rc.d/rc<x>.d/` directory, where `<x>` is the number of the runlevel.
- The following runlevels are defined by default for Red Hat Enterprise Linux:

```
# 0 - Halt
# 1 - Single user mode
# 2 - Multiuser, without NFS (without networking)
# 3 - Full multiuser mode
# 4 - Unused
# 5 - Full multi-user graphical mode (with an X-based login screen)
# 6 - reboot (Do NOT set initdefault to this)
```

- The default runlevel for the system is listed in `/etc/inittab` :

```
id:5:initdefault:
```

- The SysV init runlevel system provides a standard process for controlling which programs `init` launches or halts when initializing a runlevel.
- The configuration files for SysV init are located in the `/etc/rc.d/` directory. Within this directory, are the `rc`, `rc.local`, `rc.sysinit`, and, optionally, the `rc.serial` scripts as well as the following directories:

```
init.d/
rc0.d/
rc1.d/
rc2.d/
rc3.d/
rc4.d/
rc5.d/
rc6.d/
```

The `init.d/` directory contain the scripts used by the `/sbin/init` command when controlling services. Each of the numbered directories represents the six default runlevels configured by default under Red Hat Enterprise Linux.

5. User and Group Administration

1. Types of Users

Super User:

The root user is the super user, it is created by default.. Root user is the administrator of the system. Root user has the rights to access any file in the file system.

Regular Users:

Regular users are the normal users created by the root. They have the rights to access and execute the files present in their corresponding directories only.

System Users:

System users are nologin users. i.e. those users can't be login in the shell. Those users are created by default having uid below 500 for running and controlling various daemons.

User Administration involves

- Creating User Account
- Modifying User Account
- Deleting User Account

Likewise Group Administration also involves those.

2. Creating User Account

The command to create user account is **useradd.**

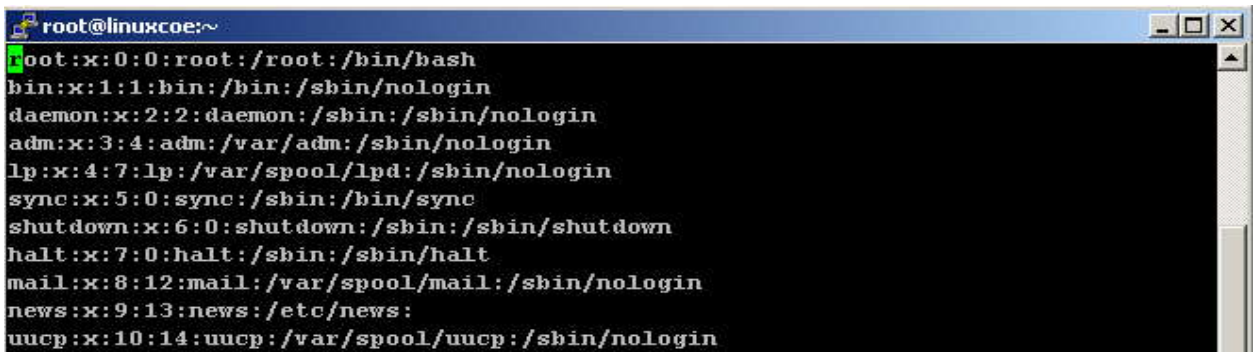
For Ex: useradd user1

The user account "user1" is now created. The user account record will be in the file /etc/passwd as below:

user1:x:500:500::/home/user1:/bin/bash

Name of the user is user1, x indicates encrypted passwd is stored in /etc/shadow file, uid of user1 is 500, primary group id is 500, home directory is /home/user1 and default login shell is bash.

Here is the sample /etc/passwd file



Here there are 7 columns,

Root - user name
x - passwd (encrypted passwd is stored in /etc/shadow file)
0 - user id
0 - primary group id
Root - comments for the user
/root - home directory for the user
/bin/bash - shell for the user

Different shells used in Linux are sh, bash, ksh and csh. For sh, bash and ksh, regular user prompt will be $ and root prompt will be #. For c shell, regular user prompt is % and root prompt is #.

When we create any user account these default entries will be entered in /etc/passwd file. The files located in /etc/skel directory will be copied to the new home directory of the user. We can assign those options while adding the user as follows:

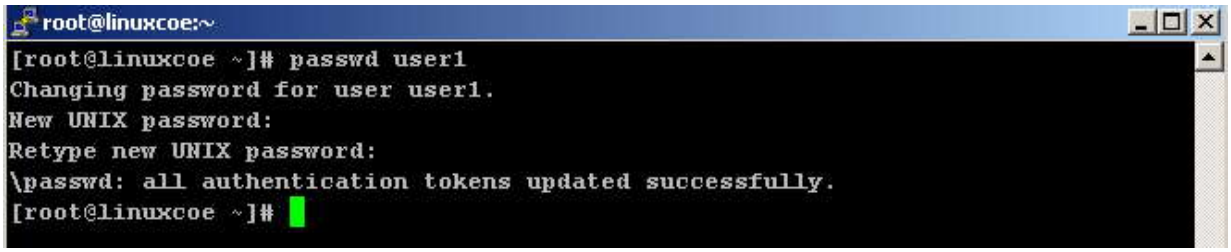
useradd –u uid –g user1 –G users,users1 –d /home/homedirectory -m –s /bin/bash –c “user1” user1

Where -u uid is the user identification number of the user account; next available uid will be assigned if not specified
-g user1 is primary group of the user
-G user1 belongs to other groups users, users1
-s /bin/bash is the shell for user1.

3. Setting password

The command passwd will be used to set password for any user. The same command will change the password of this user, if the password has been set already.

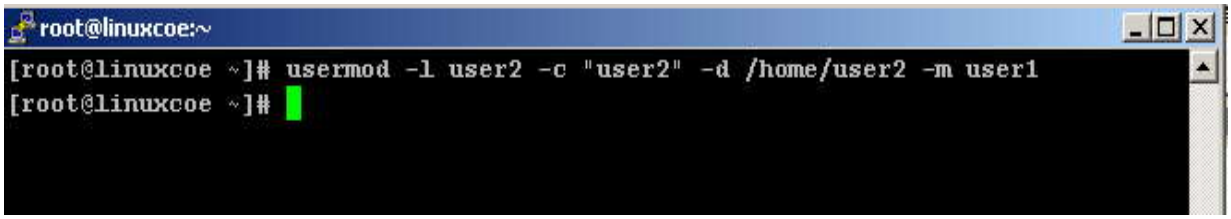
passwd user1 (username)



To remove password for user, use the command passwd –d username.

4. Modifying the user account

The command used to modify user parameters is usermod. To modify the user account user1 to user2 by changing the home directory to /home/user2, changing the shell to /bin/tcsh, use usermod command.



Where –l user2 is changing the username user1 to user2
-d /home/user2 –m is changing the home directory to user2. –m is used with conjuction to
-d to copy the contents of /home/user1 to /home/user2.

Adding the users to any group can also be done using usermod command.

To create a user named jane with a default group of finance,

#useradd jane -g finance

If existing users need to be placed in the finance group, you could run the following command:

#chgrp jim finance

If you would like to make a particular user a member of more than one group at a time (secondary groups), use the -G option with the useradd command:

#useradd joe -G wheel, finance

Note: User can be a member of only one primary group and as many number of secondary groups

The passwd command can change a user's password, as in:

#passwd joe

The Linux standard reserves the UID range from 0 through 99 for the system itself, and the range 100 through 499 for special system users (such as services and applications). RedHat LINUX thus starts regular user ID values at 500 (and ends at 60000, by default; the maximum upper limit is 65535).

To change a user's UID, run the command:
usermod -u -<UID> <username>

The following command changes jim's UID to 555:
usermod -u 555 jim

To change a user's GID, run the command:
usermod -g -<GID> <username>

The following command changes jim's GID to 555:
usermod -g 555 jim

Table 4.1. SLES Default System User Settings

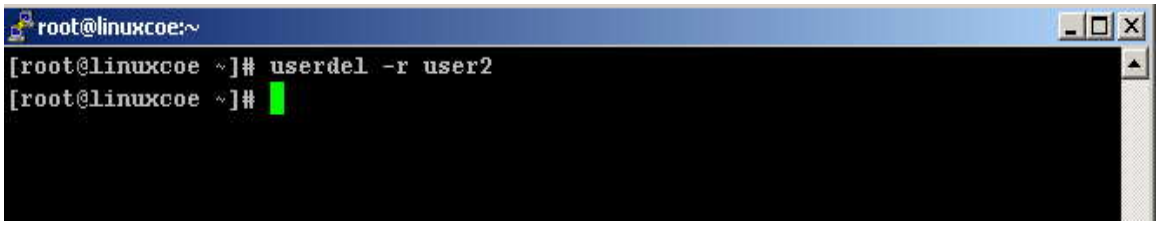
USERNAME	UID	PRIMARY GID	DESCRIPTION
root	0	0	The root (super) user
bin	1	1	Used by system services, such as lpd
daemon	2	2	Background service
lp	4	7	Printing daemon
mail	8	12	Mailer daemon
news	9	13	News system
uucp	10	14	Unix-to-Unix CoPy system
games	12	100	Games account
man	13	62	Manual pages viewer
at	25	25	Batch jobs daemon
ftp	40	49	FTP account FTP server daemon
named	44	44	Name server (DNS) daemon
mysql	60	2	MySQL database admin
pop	67	100	POP admin
sshd	71	65	SSH daemon
Mailman	72	67	GNU mailing list manager
Snort	73	68	Snort network monitor
Ntp	74	65534	NTP (Network Time Protocol) daemon
Ldap	76	70	User for OpenLDAP
nobody	65534	65533	Special user for assigning NFS permissions

5. Deleting user account

Command: **userdel**

#userdel user2.

This deletes the user name user2. But the home directory will be deleted. Supplying -r option with the command will delete the home directory also.



6. Permissions:

Linux always associates a file or directory with a user and a group. For example, assume the following file named file.txt in your home directory. If you run the command ls -l on webmaster.txt, you will get the following result:

-rw-rw-r- 1 jim webmaster 1024 Feb 21 15:10 webmaster.txt

There are three sets of permissions that every UNIX or Linux file system uses: the file's owner, the file's group, and everyone else (commonly referred to as other). The above example output is the listing of file access permissions that have been set for the file's owner, the file's group, and everyone else. This file is owned by a user named jim, the file's group is webmaster, the file size is 1,024 bytes, the file was last modified on Feb 21, at 15:10 (3:10 P.M.), and the filename is webmaster.txt.

The area of this output we're really concerned with is the grouping at the beginning of the line:

-rwx-rw-r-

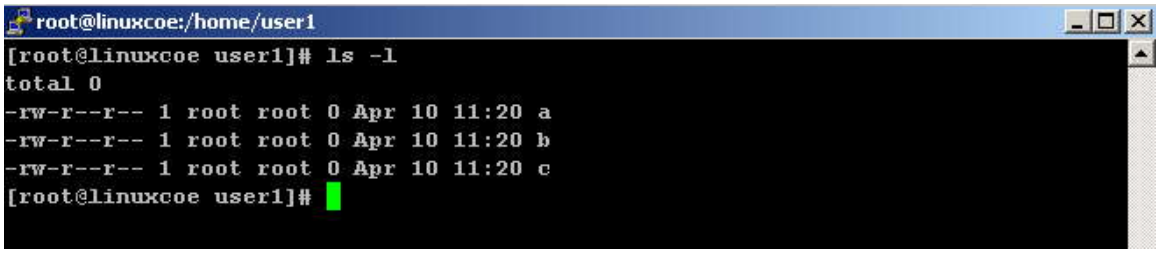
This is the area where the access permissions for the file are displayed. The first entry is a dash [-], which indicates that this is a regular file (not a directory or link). If the first charater is 'd' this indicates that this is a directory, an 'l' in this position indicates that this is a symbolic link file, 'c' indicates that this is a charater device file and 'b' indicates that this is a block devicve file. The letters r, w, and x stand for read, write, and execute.

The first entry is a dash, so we know that this file is regular file. The first triplet of letters indicates the permissions for the file's owners and contains the letters rwx. This means that the owner of the file has read, write, and execute permissions for this file. The owner can do anything to this file, including changing the access permissions. (The only other user who can do this is the superuser, or root.)

The next triplet indicates the access rights for the file's group. In our webmaster.txt example, the permissions are read and write (rw-). The users who belong to the webmaster group can read and make changes to this file, but they cannot execute the file—assuming it's an executable file such as a Perl script or a shell script.

The third triplet concerns all users other than the file's owner and all users who don't belong to the file's (webmaster group) group. In this case, the rights for "other" are read-only (r--), which means that any user other tahn jim and, or who doesn't belong to the webmaster group, can only read the file. If these users run the cat or ls command on the file, they will see that it exists and they can read the file.

Any file has default permission of 644 (rw-r--r--) and directory has 755 (rwxr-xr-x). For ex: The permission 4 indicates – Read permission, 2 indicates write permission and 1 indicates execute permission.



7. To change the permission of the file use

Each permission (read, write, and execute) is assigned a value based on a power of two, from right to left. The first permission, execute, has a value of two to the zero power, or one. The second permission, write, has a value of two to the first power, or two. The third permission, read, has a value of two to the second power, or four. The highest access right is seven, or read, write, and execute. The lowest access right is zero, or no rights.

-rw-rw-r 1 <username> <username> 1024 Feb 22 10:20 myfile.txt

In the above example the owner has read/write permission (4 + 2), or a value of six. The file's group also has read/write permission, also a value of six. Everyone else has read-only permission, or a value of four.

Now let's change the permissions on myfile.txt. To assign permissions, add the values of the permissions you want to assign, and then assign them using the chmod command.

As the owner of the file, you want execute permission. Execute has a value of four, so you'll run the command:

#chmod 764 myfile.txt

Now run ls -l on myfile.txt, and you should see something similar to this:

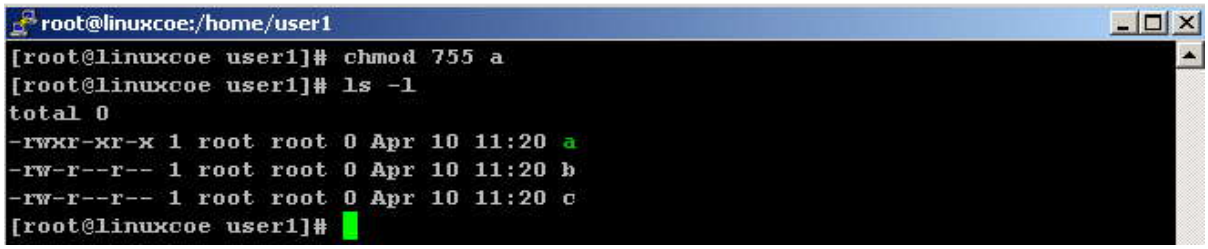
-rwx-rw-r-- 1 <username> <groupname> <size> <date> <time> myfile.txt

If you aren't happy with the file's group having write access, run the chmod command again:

#chmod 744 myfile.txt

Run ls -l myfile.txt, and you'll see that the owner has read, write, and execute access but that the file's group and everyone else has read access only.

-rwx-rw-r-- 1 <username> <groupname> <size> <date> <time> myfile.txt



Here the file permission for the file ‘a’ changed to 755.

8. **To change the ownership for a file use**

Only the superuser, or root, can change ownership of a file. This is a built -in security feature of Linux. Suppose I wrote a program that changed the access of critical system files that only root had access to on a server. I could then simply change the ownership of the program file to root. When I ran the program, I would have root access to all files affected by the program.

To change file or directory ownership, login as root, or su to root, and then run the com mand:

chown <username> <filename>

To change ownership of myfile.txt to a user named bill, run:

#chown bill myfile.txt

To change the file's group with the same command, run:

chgrp <new group> <filename>

To change the group for myfile.txt to admin, run:

#chgrp admin myfile.txt

To change the owner and group of a file in one step, run:

#chown bill:admin myfile.txt

To change the ownership or group of all files and directories under a parent directory, use the -R option with the chown command. If you want to change the owner and group of all files and directories under the /www directory to user jim and group internet, run the command:

#chown -R jim.internet /www

If you only want to change the file's group, run the command:

chgrp <new group> <filename>

To change the group of all text files in the /www directory to the group jim, run the command:

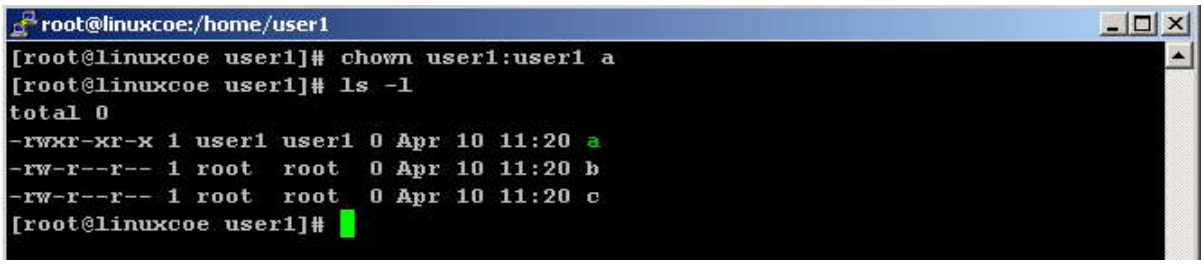
#chgrp -R jim /www

The chown and chgrp commands may also be used with an asterisk (*) to change the permissions or group of all files in a directory. For example, type cd /www to change to the /www directory. Then, to change the permissions on all text files in the /www directory, run:

#chmod 755 *txt

To change the group for all files with the extension pl in the /www directory, run:

#chgrp internet *pl



Here the file owner changed to user1 and group owner changed to user1 instead of root. We can also use chgrp command to change the group owner of the file.

9. Setting disk quota to users

In order to manage disk space effectively, you need to restrict the amount of disk space used on each partition by each user or group of users as your disk drives become filled with data. This can be achieved setting up disk quota in RedHat Linux.

Steps to setup disak quota:

1. Inform all to logout

Use the who command to see which users are logged in. If there are any, besides yourself, send a message stating that the system is about to shutdown with the wall command:

```
[root@bigboy tmp]# who
root     pts/0        Nov  6 14:46 (192-168-1-242.my-site.com)
bob      pts/0        Nov  6 12:01 (192-168-1-248.my-site.com)
bunny    pts/0        Nov  6 16:25 (192-168-1-250.my-site.com)
[root@bigboy tmp]# wall The system is shutting down now!

        Broadcast message from root (pts/0) (Sun Nov  7 15:04:27 2004):

        The system is shutting down now!
        [root@bigboy tmp]#
```

2. Enter single user mode.

```
[root@bigboy tmp]# init 1
```

3. Edit Your /etc/fstab File

The /etc/fstab file lists all the partitions that need to be auto-mounted when the system boots. You have to alert Linux that quotas are enabled on the filesystem by editing the /etc/fstab file and modifying the options for the /home directory. You'll need to add the usrquota option. In case you forget the name, the usrquota option is mentioned in the fstab man pages.

The old /etc/fstab looked like

```
LABEL=/home      /home         ext3    defaults          1 2
```

but your new /etc/fstab should be

```
LABEL=/home      /home         ext3    defaults,usrquota  1 2
```

4. Remount The Filesystem

Editing the /etc/fstab file isn't enough, Linux needs to reread the file to get its instructions for /home. You can do this using the mount command with the -o remount qualifier.

```
bash-2.05b# mount -o remount /home
```

5. Get Out of Single-user Mode

Return to your original run state by using either the exit, init 3 or init 5 commands. Continue to the next step once the system is back to its normal state. You can also use the exit command to return to your default runlevel.

```
bash-2.05b# exit
```

6. Create The Partition Quota Configuration Files

The uppermost directory of the filesystem needs to have an aquota.user file (defines quotas by user) and an aquota.group file (defines quotas by group), or both. The man page for quota lists them at the bottom. In this case just enable per-user quotas for the /home filesystem.

```
[root@bigboy tmp]# touch /home/aquota.user

[root@bigboy tmp]# chmod 600 /home/aquota.user
[root@bigboy tmp]#
```

7. Initialize The Quota Table

Editing the /etc/fstab file and remounting the file system only alerted Linux to the fact that the filesystem has quota capabilities. You have to generate a quota table, separate from the aquota files, that lists all the current allocations for each user on the file system. This table will be automatically and transparently updated each time a file is modified. Linux compares the values in this table with the quota limitations that the systems administrator has placed in the aquota files and uses this information to determine whether the user has rights to increased disk usage. You initialize the table with the quotacheck command. Be prepared: You'll get an error the first time you enter the command, because Linux will realize that the aquota file wasn't created using one of the quota commands:

```
[root@bigboy tmp]# quotacheck -vagum
quotacheck: WARNING -  Quotafile /home/aquota.user was probably truncated.
Can't save quota settings...
quotacheck: Scanning /dev/hda3 [/home] done
quotacheck: Checked 185 directories and 926 files
[root@bigboy tmp]#
```

8. Edit The User's Quota Information

Now you need to edit the user's quota information. The edquota command enables you to selectively edit a portion of the aquota.user file on a per-user basis:

```
[root@bigboy tmp]# edquota -u mp3user
```

The command will invoke the vi editor.

Disk quotas for user mp3user (uid 503):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hda3	24	0	0	7	0	0

From here, you can edit a number of fields:

Blocks: The amount of space in 1K blocks the user is currently using.
Inodes: The number of files the user is currently using.
Soft Limit: The maximum blocks/inodes a quota user may have on a partition. The role of a soft limit changes if grace periods are used. When this occurs, the user is only warned that their soft limit has been exceeded. When the grace period expires, the user is barred from using additional disk space or files. When set to zero, limits are disabled.
Hard Limit: The maximum blocks/inodes a quota user may have on a partition when a grace period is set. Users may exceed a soft limit, but they can never exceed their hard limit.
Here user mp3user is limited to a maximum of 5MB of data storage on /dev/hda3 (/home):

Disk quotas for user mp3user (uid 503):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hda3	24	5000	0	7	0	0

9. Testing

Linux checks the total amount of disk space a user uses each time a file is accessed and compares it against the values in the quota file. If the values are exceeded, depending on the configuration, then Linux prevents the creation of new files or the expansion of existing files to use more disk space.

10. Setting local and global user profile

Local User profile:

Local user profile is a profile set for individual users. Anything configured in his local initialization file will be executed for his login alone. The file ~/.bash_profile is the personal initialization file, executed for login shells. There is also ~/.bashrc file which is the individual per-interactive-shell startup file. Common uses for ~/.bash_profile are to set environment variables such as PATH, JAVA_HOME, create aliases for shell commands and set the default permissions for newly created files etc. The file ~/.bashrc is similar, with the exception that .bash_profile runs only for Bash login shells and .bashrc runs for every new

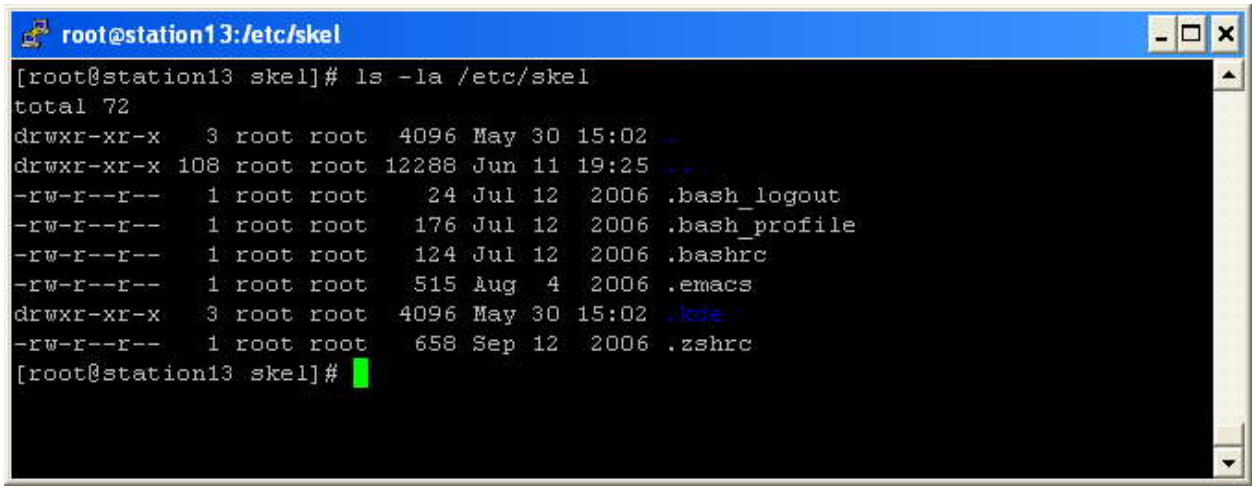
Bash shell. ~/.bashrc file runs every time you open a new non-login bash shell such as xterm / aterm, and ~/.bash_profile runs only with login shells i.e when you first log in into system.

Global user profile:

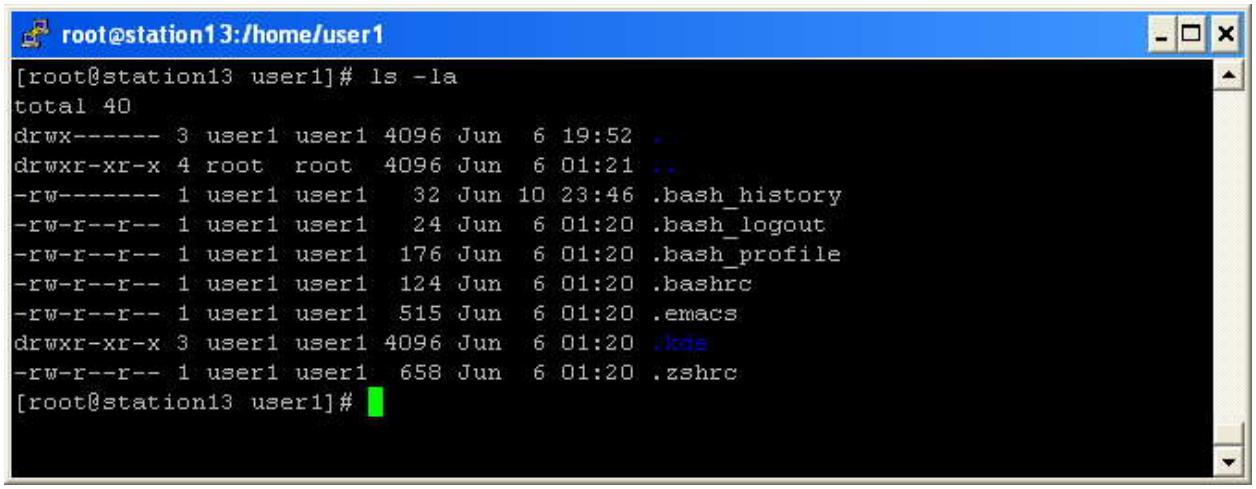
This is a profile configured for all users in a server. The system-wide global initialization file is /etc/profile. Whatever configured inside this file will be executed for all users logging in to that server. System wide environment and startup programs that need to be executed while each user is logging in to the server will be configured in the file /etc/profile.

Default user profile:

While each user is created, default user profile will be created for him. This is nothing but copying file from /etc/skel directory to his home directory. If you look at the /etc/skel directory:



The same set of files will be copied to the home directory of user1, while you create a user account user1



11. Group Administration

Same as user administration we have group administration which includes creating, modifying, deleting group accounts, adding users to group and removing users from group.

Creating Group accounts:

Syntax is

groupadd [-g gid [-o]] group

#groupadd test1

This will create a test1 group

Modifying Group account:

Syntax is

groupmod [-g gid [-o]] [-n group_name] group

#groupmod test1 test2

This will modify group name test1 to test2

Deleting Group account:

Syntax is

groupdel groupname

#groupdel test2

This will delete the test2 group

Printing the groups a user is in

Syntax is

groups [username]

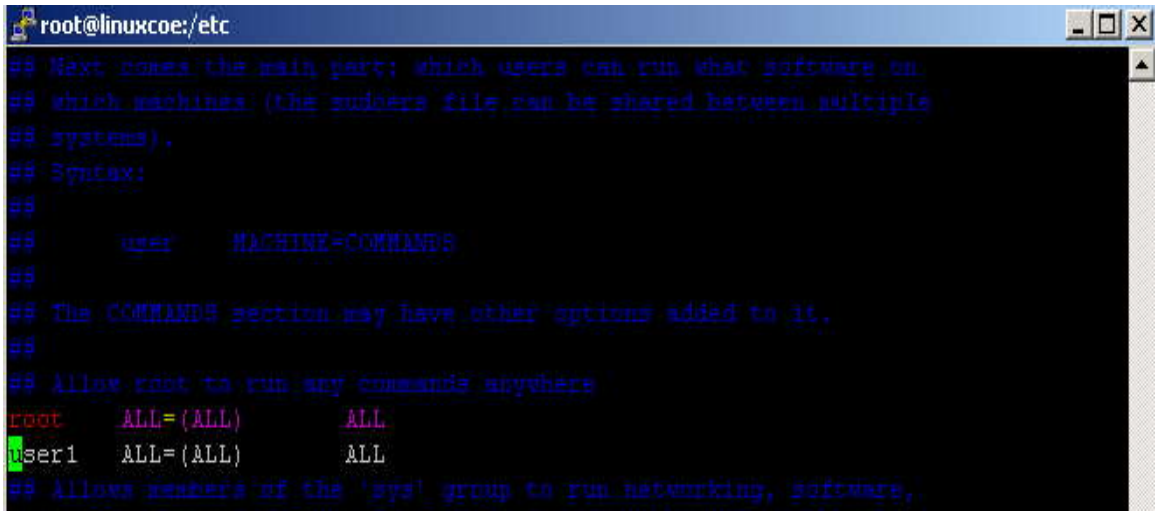
This command displays what groups the given user is a member. If no username is given, it defaults to the current user.

```
# groups
root
# groups admin
test1 : test2
```

/etc/group file contains the group account records and /etc/gshadow file contains the group passwords.

12. Giving Sudo Rights to the Users

To give root access to the regular users, add the following lines in /etc/s udoers file.



We can login as root user, when we logged as regular user using the following command

$sudo su

It will prompt you for root passwd, if you type the root password, you will have root access then.

6. Device Management and File System

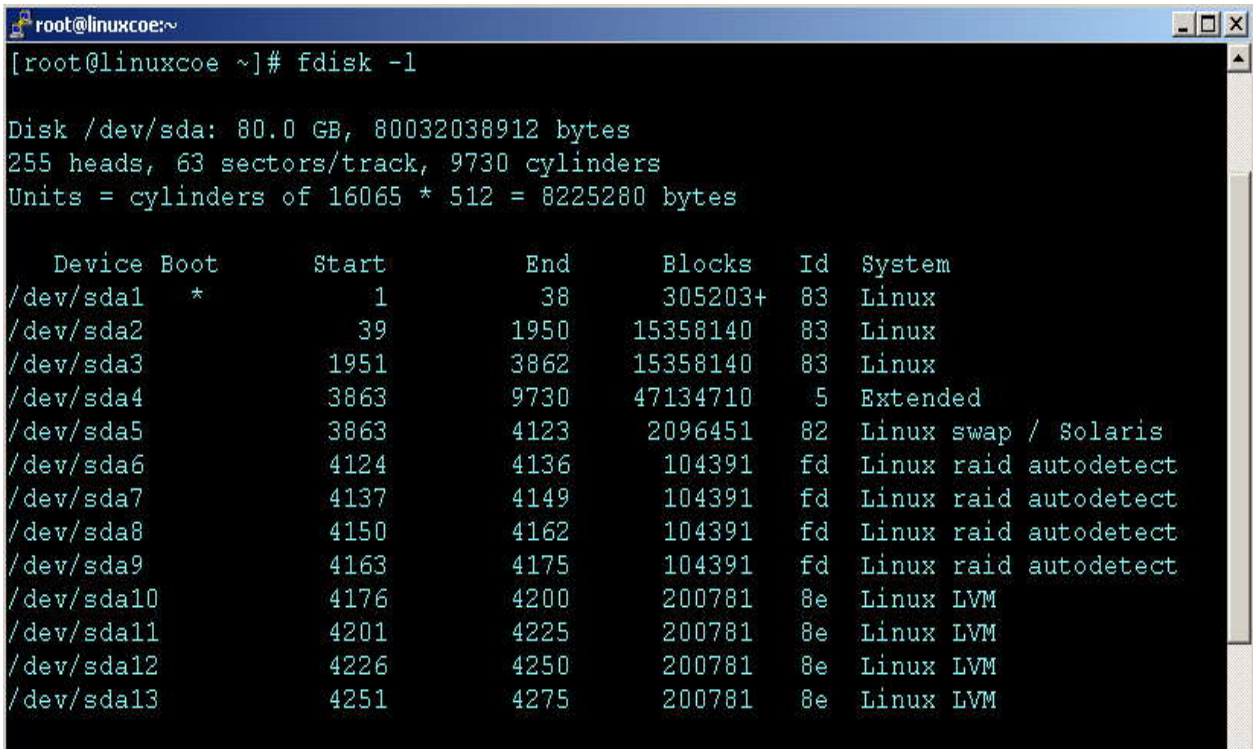
File system in Linux are organized in hierarchy, beginning from root (/) and continuing downward in structure of directories and subdirectories. In Linux local disk, network file system, CD-ROM and any storage medium fits neatly into the directory structure.

1. Subdirectories of the root directory

Directory	Content
/bin	Common programs, shared by the system, the system administrator and the users.
/boot	The startup files and the kernel, vmlinuz.
/dev	Contains references to all the CPU peripheral hardware, which are represented as files.
/etc	Most important system configuration files.
/home	Home directories of the common users.
/initrd	(on some distributions) Information for booting.
/lib	Library files, includes files for all kinds of programs needed by the system and the users.
/lost+found	Every partition has a lost+found in its upper directory. Files that were saved during failures are here.
/misc	For miscellaneous purposes.
/mnt	Standard mount point for external file systems or devices.
/net	Standard mount point for entire remote file systems
/opt	Typically contains extra and third party software.
/proc	A virtual file system containing information about system resources.
/root	The administrative user's home directory.
/sbin	Programs for use by the system and the system administrator.
/tmp	Temporary space for use by the system, cleaned upon reboot.
/usr	Programs, libraries, documentation etc. for all user-related programs.
/var	Storage for all variable files and temporary files created by users, such as log files, the mail queue, the print spooler area.

For hard drive partitions, create a mount point in the file system and then connect the disk to that point in the file system. For removable media (such as CD, DVD, USB flash drives) mount points are automatically created and connected in the /media directory when those items are connected.

To view what partitions is currently setup on the hard disk, use fdisk command.



Here is some supported file system for Linux operating system. The default file system in latest versions of Linux is ext3.

Filesystem	Description
bfs	Boot File System—a small bootable filesystem used to hold the files necessary for system startup; it is commonly used on UNIX systems
cdfs	Compact disc filesystem—used to view all tracks and data on a CD-ROM as normal files
ext2	Second extended filesystem—currently the most common filesystem used on Linux, it supports Access Control Lists (individual user permissions). It retains its name from being the new version of the original extended filesystem, based on the Minix filesystem
ext3	Third extended filesystem; a variation on ext2 that allows for journaling and thus has a faster startup and recovery time
hfs	Hierarchical File System—a filesystem native to Apple Macintosh computers
hpfs	High Performance File System—an IBM-proprietary OS/2 filesystem that provides long file name support and is optimized to manipulate data on large disk volumes
iso9660	The CD-ROM filesystem—originated from the International Standards Organization recommendation 9660 and used to access data stored on CD-ROMs
minix	The MINIX filesystem—the filesystem used by Linus Torvalds in the early days of

2. *Device files associated with different devices*

Everything is referred as a file in Linux, so devices connected to the Linux system will also be referred as a file. Here are few device files associated with different devices.

Device File	Description	Block or Character
/dev/fd0	First floppy disk on the system	Block
/dev/fd1	Second floppy disk on the system	Block
/dev/hda1	First primary partition on the first IDE hard disk drive (primary master)	Block
/dev/hdb1	First primary partition on the second IDE hard disk drive (primary slave)	Block
/dev/hdc1	First primary partition on the third IDE hard disk drive (secondary master)	Block

Device File	Description	Block or Character
/dev/hdd1	First primary partition on the fourth IDE hard disk drive (secondary slave)	Block
/dev/sda1	First primary partition on the first SCSI hard disk drive	Block
/dev/sdb1	First primary partition on the second SCSI hard disk drive	Block
/dev/tty1	First local terminal on the system ([Ctrl]-[Alt]-F1)	Character
/dev/tty2	Second local terminal on the system ([Ctrl]-[Alt]-F2)	Character
/dev/ttyS0	First serial port on the system (COM1)	Character
/dev/ttyS1	Second serial port on the system (COM2)	Character
/dev/psaux	PS/2 mouse port	Character
/dev/lp0	First parallel port on the system (LPT1)	Character
/dev/null	A device file that represents nothing; any data sent to this device is discarded	Character
/dev/st0	The first SCSI tape device in the system	Character
/dev/usb/*	USB device files	Character

3. What is mounting and why?

As you know, you can store your data in different physical storage devices, like floppies, CD-ROMs, and hard disk drives.

In Microsoft Windows, you're probably used to accessing all your filesystems very easily: you just boot up your computer, go to *My Computer*, and find all your Windows partitions will be there. However, this isn't the case in Linux.

You're probably a bit confused at first: you put your floppy or CD into the drive and start wondering why you're not able to access it.This is because your floppies, CDs, hard disk partitions, and other storage devices must be attached to some existing directory on your system before they can be accessed. This attaching is called *mounting*, and the directory where the device is attached is called a *mount point*.

After the device is mounted, you can access the files on that device by accessing the directory where the device is attached. When you're done and want to remove the floppy or CD or other device, you need to detach, *unmount*, it before removing it.

4. How to mount

Mounting can be done by mount command.

5. Mounting Floppy

mount /dev/fd0 /mnt/floppy

To unmount the device, umount command is used (it is umount, not unmount)

umount /mnt/floppy (or) umount /dev/fd0

Here the device floppy is mounted in the mount point /mnt/floppy. Like wise we can mount other removable medias. But when we restart the system again, the removable media we have mounted no more present there.

6. Mounting USB Media

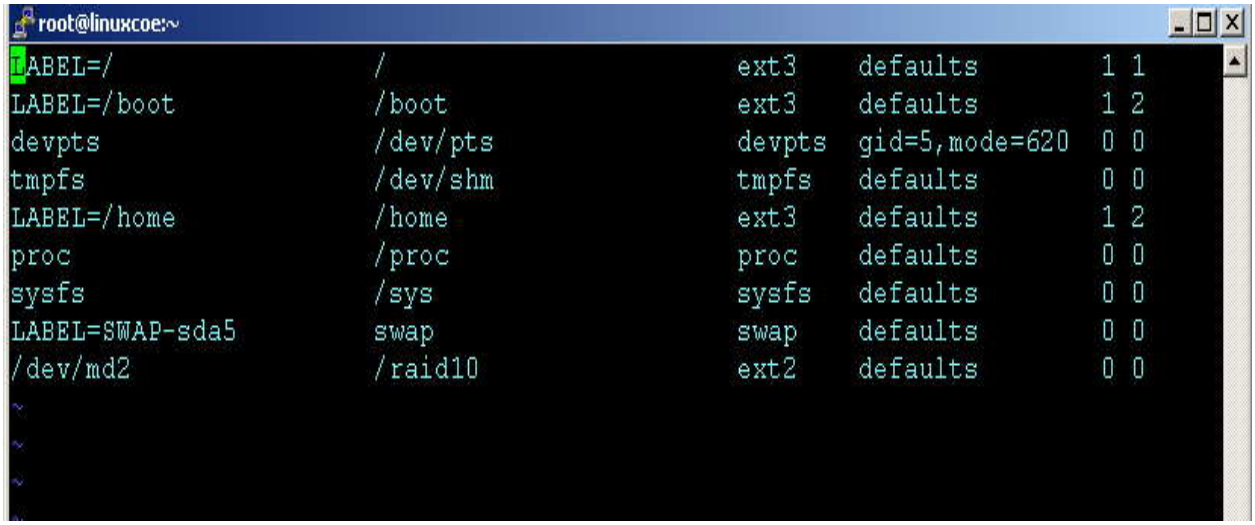
It is detected as scsi device as /dev/sdaX. We can mount it in any mount point.

7. Mounting CD/DVD

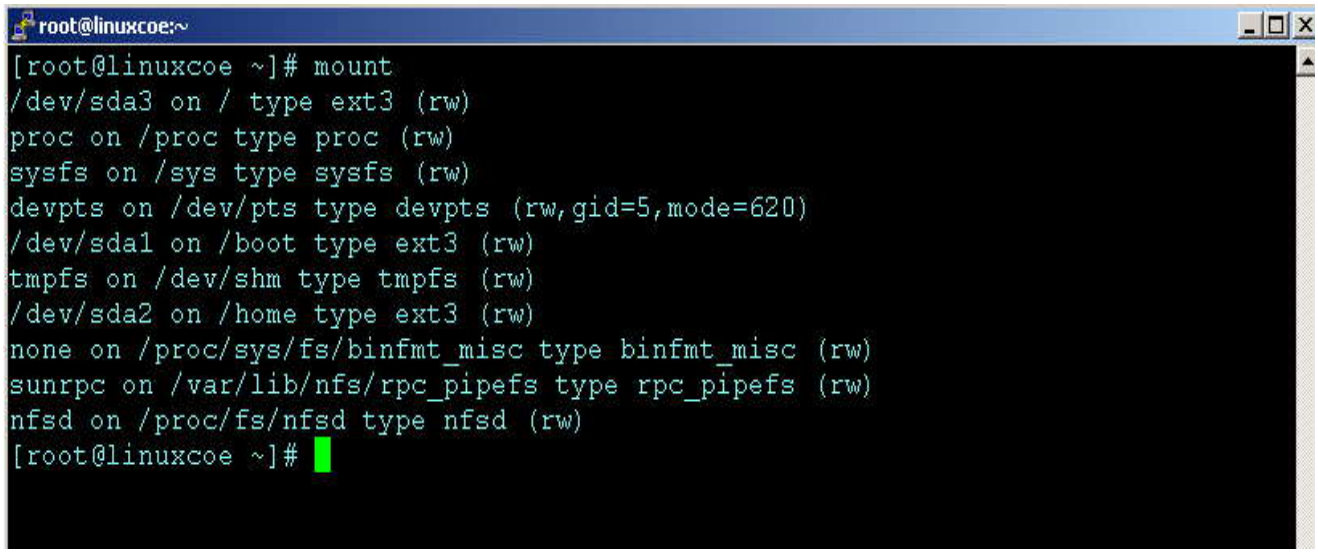
mount /media/cdrom

mount /media/cdrecorder. Eject is the command to eject CD-ROM.

To mount the devices permanently during the system boot, we have to add in the /etc/fstab file.



The file /etc/mtab contains the devices which are all currently mounted. This can be viewed using mount command.



We can pass arguments in mount command for specifying the file system we are mounting

For ex:

mount -t msdos /dev/fd0 /mnt/floppy where -t specifies the file system type and msdos is the file system for the floppy.

8. Creating a file system

It is possible to create a file system, for any supported file system type on a disk or partition that we choose. This is done with mkfs command.

mkfs -t ext3 /dev/fd0. There are also some other commands to create a file system

mke2fs /dev/fd0

9. Formatting a floppy

fdformat /dev/fd0

mformat a:

10. Adding a hard disk and creating partition in it:

The steps for adding a hard disk and creating partition are as follows:

1. Install the hard disk drive.
2. Identify the partitions or create new partitions on the new hard disk.
3. Create the file systems on the new hard disk.
4. Mount the file systems.

Let us see the above in detail:

- Install the hard disk into the computer.
- Boot the computer in Linux.
- Determine the device name for the hard disk. As root user from shell, type

dmesg | less

- Use the fdisk command to create partitions on the new disk.

Type fdisk /dev/sda or /dev/sdb according to the first or second hard disk drive, then type m for help

- To create a new partition, type n
- To choose primary partition, type p

If your are creating first partition, type 1, if second type 2.

It will ask you to enter the cylinder number, type enter to take the default.

- Type w to write changes to the hard disk. Then type q to quit from the menu.

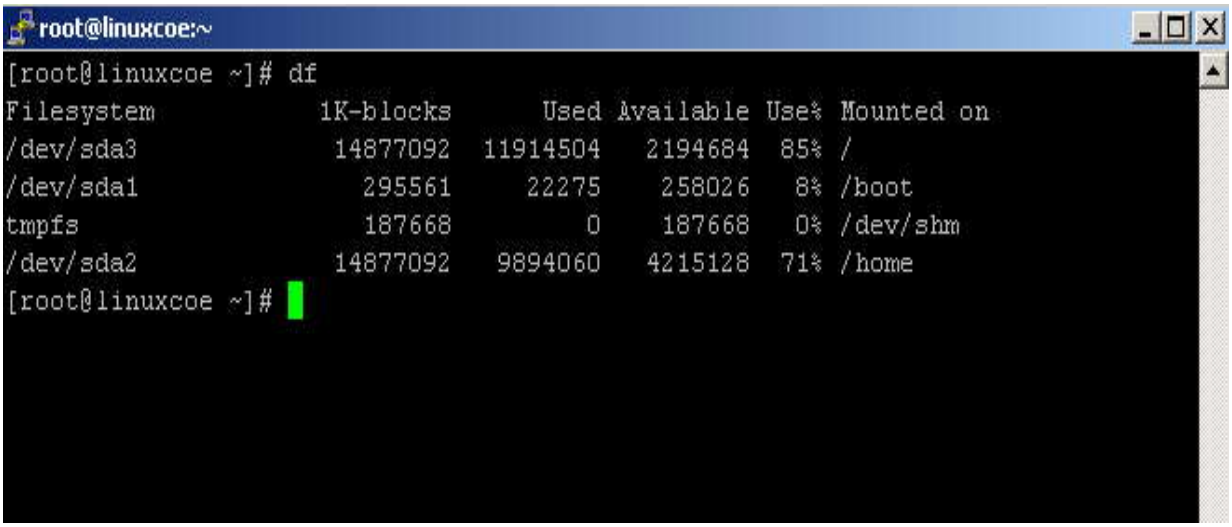
Then type partprobe to OS partition table changes.
Then we have to make file system. Use mkfs command, by default it will create ext2 filesystem, to create ext3 file system, type as follows

- mkfs –t ext3 /dev/sdb1

Mount the partition in any directory and to mount it permanently add it in /etc/fstab file.

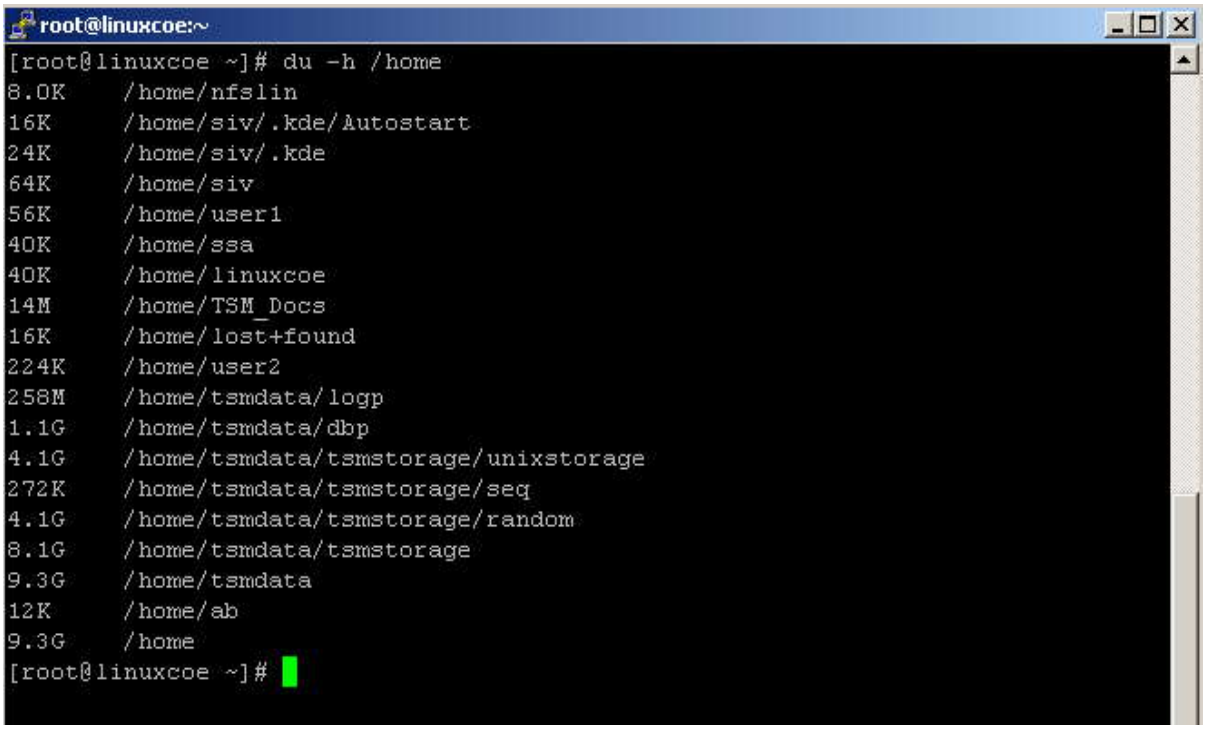
11. Checking system space

We can display the space available in the file systemes using the df command: To see the amount of space available on all of the mounted file systems on Linux computer, type df.



13. Checking Disk Usage

To find out how much space is being consumed by a particular directory, we can use du command.



14. System Performance can be monitored using the following command:

top

15. Swap Partition

Swap partition in Linux provides virtual memory space to run the larger programs. We can create a swap partition, during installation itself. In case, if we like to add space with the existing swap space, we can do it as follows.

To display the swap space enabled on your Linux system, run the command “swapon –s” as below:

[root@linuxcoe ~]# swapon -s
Filename Type Size Used Priority
/dev/sda3 partition 1052248 176 -1
[root@linuxcoe ~] #

The above output shows ‘/dev/sda3’ is now configured as swap partition.

To add swap space with this, you should create an empty file with the required size using the command “dd” as below:

```
[root@linuxcoe ~]# dd if=/dev/zero of=/swapfile bs=1024 count=65536
65536+0 records in
65536+0 records out
[root@linuxcoe ~]#
```

Configure this empty file as swapspace:

```
[root@linuxcoe ~]# mkswap /swapfile
Setting up swapspace version 1, size = 67104 kB
[root@linuxcoe ~]#
```

Enable this swap space:

```
[root@linuxcoe ~]# swapon /swapfile
```

Now, "swapon –s" displays the swap space enabled at present:

```
[root@linuxcoe ~]# swapon -s
Filename Type Size Used Priority
/dev/sda3 partition 1052248 176 -1
/swapfile file 65528 0 -3
[root@linuxcoe ~]#
```

At any time, you can disable swap space by running the command "swapoff"

```
[root@linuxcoe ~]# swapoff /swapfile
[root@linuxcoe ~]# swapon -s
Filename Type Size Used Priority
/dev/sda3 partition 1052248 176 -1
[root@linuxcoe ~]#
```

16. Checking and Repairing File System

The command used to check and repair a file system is **fsck**.

fsck is used to check and optionally repair one or more Linux file systems. File system can be a device name (dev/sda1), a mount point (/mnt/) or an ext2 label.

Normally, the fsck program will try to handle file systems on different physical disk drives in parallel to reduce the total amount of time needed to check all of the file systems.

- **Running fsck command**

Do not run fsck on a live or mounted file system. fsck is used to check and optionally repair one or more Linux file systems. Running fsck on a mounted file system can usually result in disk / data corruption. There are two ways to run fsck

- **Take down system to single user mode and unmout system**.

For ex. To run fsck command in /home(/dev/sda3) directory

#init 1

#umount /home

#fsck /home (or) fsck /dev/sda3

- **Boot from the installation CD into rescue mode**

If you are using RHEL Linux, boot from first CD and at boot prompt type linux rescue nomount:

boot: linux rescue nomount

Now make new node for disk and partition 3:
mknod /dev/sda
mknod /dev/sda3
fsck /dev/sda3

Don't forget to reboot the system
exit
reboot

- **Running fsck for root file system**

Log into the system as single user and unmount the root file system and then run fsck command. After file system check is completed reboot the system.

17. Setting up LVM

Linux provides a flexibility of viewing more than one physical hard disk as a single partition. Therefore, it is easy for administrators in allocating storage to applications and users.

Today we are going to about creating Logical Volumes in Linux.

Here are the steps:

Step 1: Before using a disk in a volume group we have to prepare it, by running "pvcreate" command.

Run pvcreate on the disks
pvcreate /dev/sda
pvcreate /dev/sdb
pvcreate /dev/sdc

This will destroy any data on /dev/sda, /dev/sdb, and /dev/sdc. Running the above commands creates a volume group descriptor area (VGDA) at the start of the disks.

Step 2: Create a volume group

vgcreate my_volume_group /dev/sda /dev/sdb /dev/sdc/

Step 3: Run vgdisplay to verify volume group

```
# vgdisplay
--- Volume Group ---
VG Name       my_volume_group
VG Access         read/write
VG Status         available/resizable
VG #              1
MAX LV              256
Cur LV            0
Open LV           0
MAX LV Size       255.99 GB
Max PV            256
Cur PV            3
Act PV            3
VG Size           1.45 GB
PE Size           4 MB
Total PE          372
Alloc PE / Size   0 / 0
Free  PE / Size   372/ 1.45 GB
VG UUID           nP2PY5-5TOS-hLx0-FDu0-2a6N-f37x-0BME0Y
```

The most important things to verify are that the first three lines are correct and that the VG Size item is the proper size for the amount of space in all four of your disks.

Step 4: Creating the Logical Volume

If the volume group looks correct, the we need to create a logical volume on top of the volume group. In the below example we will create just a single logical volume of size 1GB on the volume group.

```
# lvcreate -L1G -nmy_logical_volume my_volume_group
lvcreate -- doing automatic backup of "my_volume_group"
lvcreate -- logical volume "/dev/my_volume_group/my_logical_volume" successfully created
```

Step 5: Create the File System

Create an ext2 file system on the logical volume

```
# mke2fs /dev/my_volume_group/my_logical_volume
```

Step 6: Test the File System

Mount the logical volume and check to make sure everything looks correct

```
# mount /dev/my_volume_group/my_logical_volume /mnt
# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda1              1311552    628824    616104  51% /
/dev/my_volume_group/my_logical_volume
                       1040132        20    987276   0% /mnt
```

Now Logical Volume is ready to use.

18. Configuring RAID

Multiple disks grouped together into arrays to provide better performance, redundancy or both. Raid levels RAID 0, 1 and 5 are supported. Spare disks add extra redundancy. Either partition or whole disks can be used to create software RAID devices.

Let us see the most commonly used RAID types

RAID 0 or Stripping: Two or more disks used to create a single large high performance volume. If the drives of equal size are used performance will be better. Chance of failure is very high. The major disadvantage is no data redundancy.

RAID 1 or Mirroring: This RAID level needs at least two disks which duplicate the data i.e. data updated simultaneously. Hot spare disks can be used to improve fault-tolerance. Array size equals the size of the smallest disk used.

RAID 5: This type provides rotating parity array. Three or more disks with zero or more hot spares used. Redundancy is achieved by splitting parity between all disks, one disk can be lost

without causing array failure. It is best for multi-user systems in which performance is not critical.

RAID Configuration

mdadm (Multiple Device Administration) provides administration interface to software RAID.

Before configuring RAID, the partitions need Linux RAID type. For ex. If you are configuring partition 5 and 6 in RAID 1, type as follows:

#fdisk /dev/sda
Command (m for help): t
#Partition number (1-8): 5
#Hex code (type L to list codes): L
If you type L, you can see list of codes to specify the type. Linux RAID type is **fd.** So type fd then
#fd
#wq (to save the changes and to quit)

Then in command prompt, type **partprobe** (To inform the OS partition table changes)

To create RAID of type RAID 1 for two partitions /dev/sda5 and /dev/sda6, type as follows

#mdadm –C /dev/md0 –a yes –l 1 –n 2 /dev/sda{5,6}

Here -C - to create RAID
-a yes - to create device /dev/md0
-l - specifies raid level
-n - specifies number of partitions [i.e. sda1 and sda2]

Then make file system for /dev/md0

#mke2fs –j /dev/md0
It creates ext2 file system for /dev/md0.

Then mount the device to any mount point.
To view the status of RAID Devices, we can use the command

#mdadm –D /dev/md0

Now we will check RAID by setting a partition as faulty.

#mdadm –f /dev/md0 /dev/sda5

Remove the faulty partition using the command

#mdadm -r /dev/md0 /dev/sda5

If you give the command mdadm –D /dev/md0 you will see the active partition (/dev/sda6) and removed partition (/dev/sda5) specified. But you can see the contents in the RAID device which is mounted in any mount point. It shows RAID is working properly.

For RAID 5, you can create as follows

#mdadm –C /dev/md0 –a yes –l 5 –n 3 –x 1 /dev/sda{7,8,9,10}

Where –l 5 specifies – type 5 RAID
-n 3 specifies – 3 partitions
-x 1 specifies – spare partition

Partitions 7, 8,9,10 are configured as Linux raid partitions.

7. Storage Management

The command dump command is used to backup the file system. The command restore is used to restore the backup from any backup medium.

Syntax is

```
# dump `option' `parameter' `filesystem'
```

- options
  0-9 : dump level
  B : number of records per volume
  b : blocksize per record (KB)
  h : dump level below which the nodump attribute affects
  f : output file (tape)
  d : tape density
  n : notify to the operator
  s : tape length
  u : update /etc/dumpdates
  T : specify the date to record in /etc/dumpdates
  W : print the filesystems to be dumped with marks
  w : print the filesystems which need to be dumped
- parameters
  Specify the parameters corresponding to the options in sequence.
  For example, if the option is ``sbf'', the following parameters
  should be the order:
  dump sbf `tape length' `blocksize' `filesystem' `output file'
- filesystem
  mount point or a device name of the filesystem to dump

Typical command line looks like:

```
# dump 0uf /dev/nst0 /home
```

The number as the first parameter is the dump level between 0-9. The `0' here means the full backup, and the meaning of other numbers will be discussed later. The `u' specifies that the /etc/dumpdates file will be updated so that the following incremental backup can use the record. /etc/dumpdates contains the time and the level of the dump. This will be referred to at the following invocation of dump for incremental backups or from w/W options.

If you dump to the tape, you may have to specify the blocksize and the tape length. When you encountered the situation: `the tape capacity should be enough but the tape reaches its end during the dump', you may want to modify these parameters. Ad hoc workaround is to specify the over-estimated capacity with B option.

```
# dump 0uBf 2300000 /dev/nst0 /home
```

This example specifies that the tape capacity is 2.3GB (i.e. the number of records for that; 2.3GB divided by the blocksize), which is the slightly too large estimation. The options to specify the density or the tape length are prepared to teach the dump when to exchange the media. But most recent drives can detect the tape end properly, you should have no problem this way.

When dump is invoked normally, backup proceeds with some messages printed on the console, which include the estimated duration until the backup finishes. It is a good idea to leave this session foreground rather than to send it to background, until you are convinced the things work fine. If it reaches the tape end or if some error occurs, you will be requested to do some choices in the interactive session.

When the dump session finishes properly, you can dump another filesystem if the tape capacity remains enough.

1. **Taking full backup using dump**

```
root@linuxcoe:/home/sun
[root@linuxcoe sun]# dump -0uf /home/sun/backup /boot
  DUMP: Date of this level 0 dump: Wed Mar 12 18:30:34 2008
  DUMP: Dumping /dev/sda1 (/boot) to /home/sun/backup
  DUMP: Label: /boot1
  DUMP: Writing 10 Kilobyte records
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 7064 blocks.
  DUMP: Volume 1 started with block 1 at: Wed Mar 12 18:30:34 2008
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /home/sun/backup
  DUMP: Volume 1 completed at: Wed Mar 12 18:30:35 2008
  DUMP: Volume 1 7220 blocks (7.05MB)
  DUMP: Volume 1 took 0:00:01
  DUMP: Volume 1 transfer rate: 7220 kB/s
  DUMP: 7220 blocks (7.05MB) on 1 volume(s)
  DUMP: finished in 1 seconds, throughput 7220 kBytes/sec
  DUMP: Date of this level 0 dump: Wed Mar 12 18:30:34 2008
  DUMP: Date this dump completed:  Wed Mar 12 18:30:35 2008
  DUMP: Average transfer rate: 7220 kB/s
  DUMP: DUMP IS DONE
[root@linuxcoe sun]#
```

Incremental backup can be taken daily in tapes by specifying higher number(1-9) that have been changed or added since the most recent dump of the lower dump level.

The backup behavior of dump is dependent on the specified dump level, which is an integer between 0-9. If not given, default value 1 is used. When invoked with some level, dump selects the files which have been updated since it ran with SMALLER level than the current one, and backups them.

For example, if level-5 backup is invoked after level-4 backup, the files updated after the level-4 backup will be recorded to this level-5 backup archive.

Using this, you can make incremental backups easily. If you run dump with:

Day 1 level 0

Day 2 level 1
Day 3 level 2
Day 4 level 3
Day 5 level 4
Day 6 level 5
Day 7 level 6
Day 8 level 7

It is possible to dump over the network: you can use "rdump". It takes the same arguments as dump except for a remote hostname to send the dump data (actually rdump and dump share the one executable binary). In the following example, we dump the files in `/home` directory on the local machine, to a tape device on the remote machine named "tapeserver".

```
#rdump 0uf tapeserver:/dev/nst0 /home
```

2. Restoring Full Backup

The command restore is used to extract files from the archive created by "dump", and to recover them on your target filesystem. It is possible either to recover the whole filesystem at once, or to choose the files to retrieve interactively. A piped combination of "dump" and "restore" can duplicate the contents of a filesystem on another filesystem. As rdump, there is an r-version of the restore, "rrestore", which can read the archive in the tape drive on a remote host.

Usage: restore `key' [key-modifier]
Keys (summary):
i Interactive restoration of specified files
r Extract the contents of the whole archive in the current directory
R Resume interrupted full-restoration
t List filenames on the backup archive
C Compare the contents of the archive with the current filesystem
T Specify the temporary directory
s Specify the position of the archive in the tape
x Only the named files are extracted from the archive
f Specify the archive file
h Extract only the directory when specified, rather than with its contents
v verbose output
y Do not query on error.

To retrieve all the files from the archive in tape media, and write them under the current directory:

restore rf /dev/nst0

To do interactive restoration from the 3rd backup on the tape media:

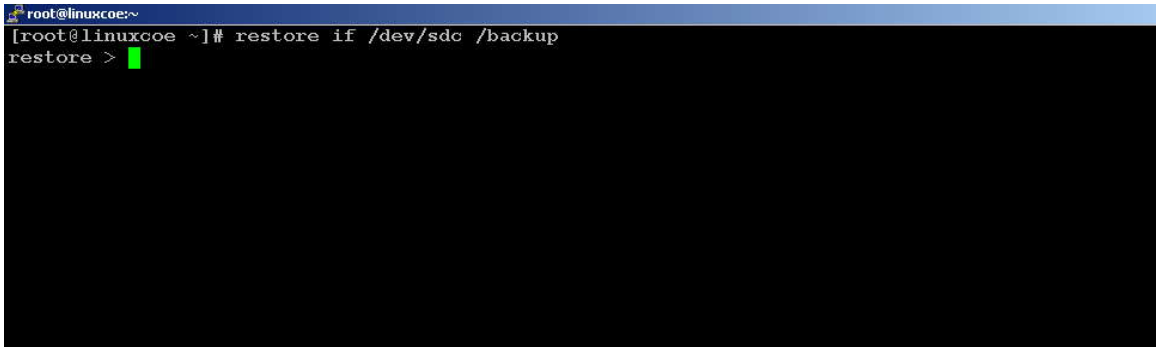
restore isf 3 /dev/nst0

To list filenames in the archive stored in the tape of the host `tapeserver' :
rrestore tf tapeserver:/dev/nst0

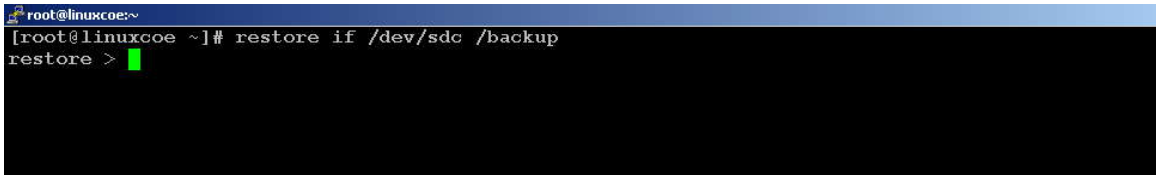
3. Restoring an Individual file

Individual files can be recovered by interactive mode.
To restore files from dump in interactive mode, use the following steps.

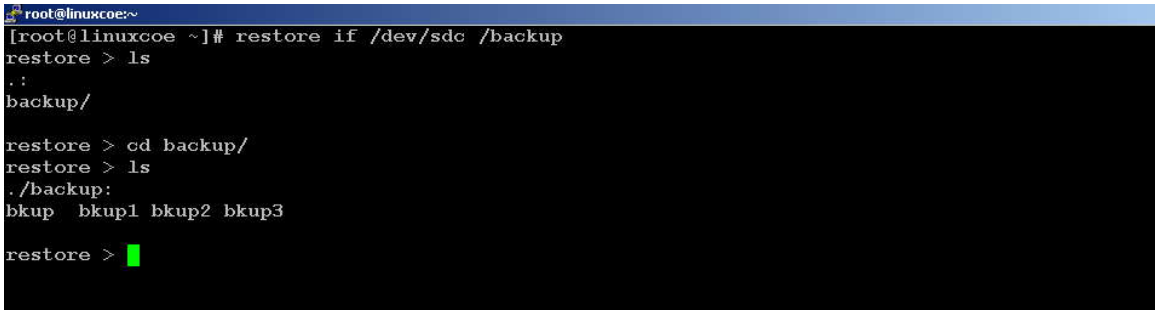
- **restore -if /dev/sdc /backup**



Once you execute the command, you will be in restore prompt.



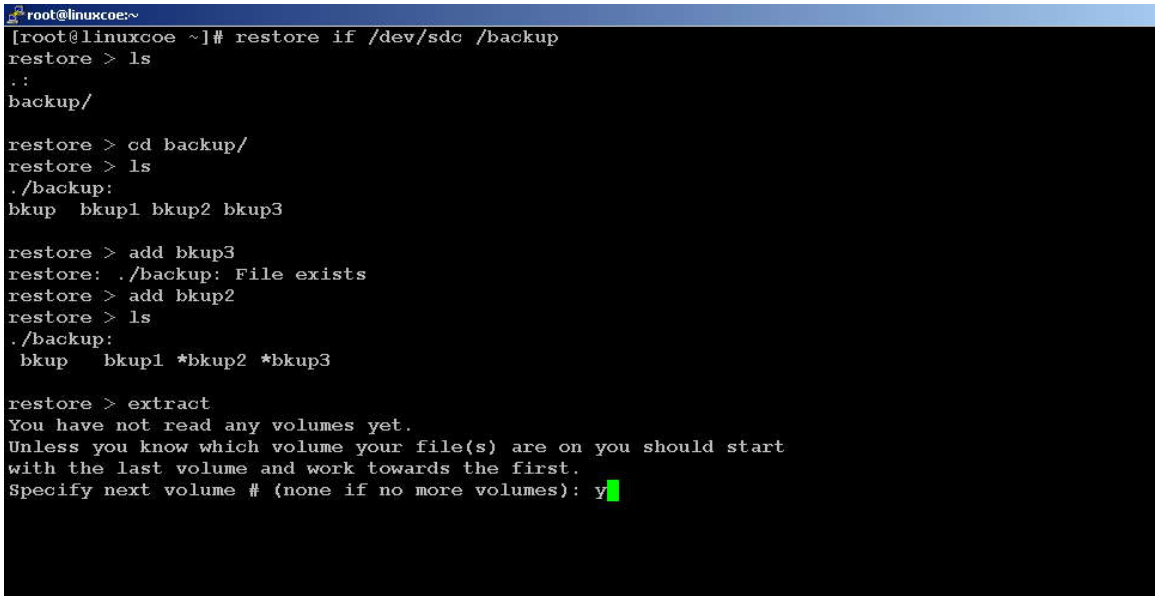
- **Type ls to view the contents in backup medium**



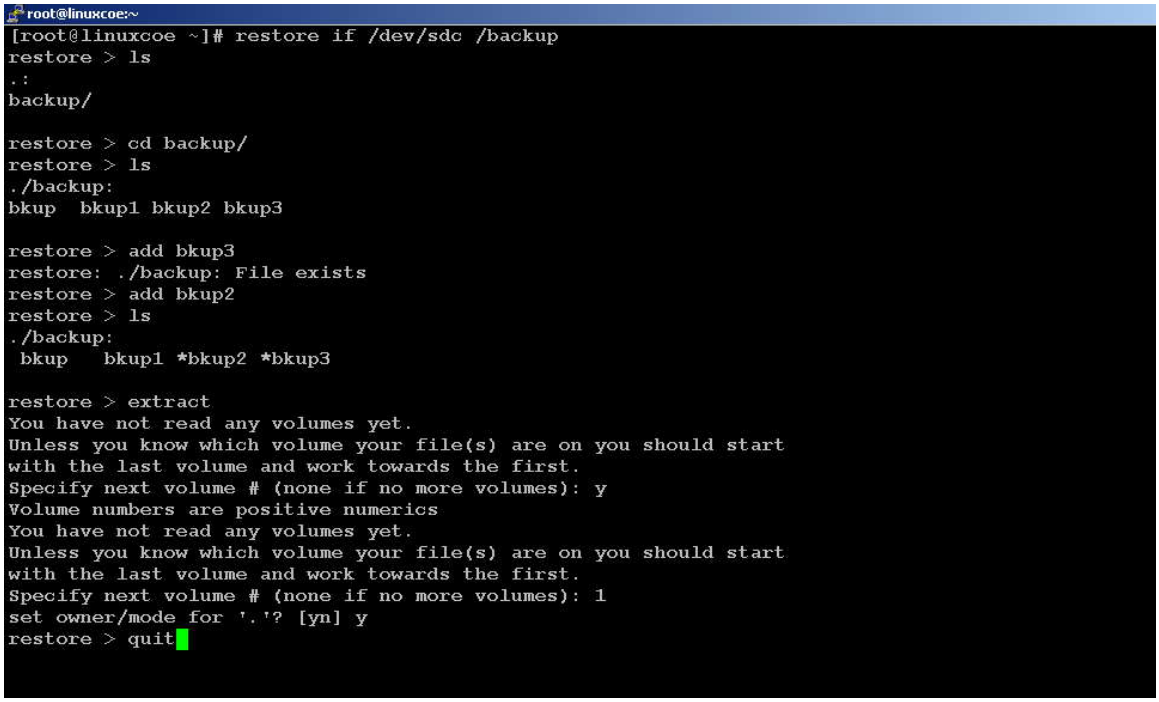
- **Type add to select the files to extract.**

If you type again **ls** command, the files chosen to extract are marked with asterisk* character.

- **Then type extract to extract the files.**



- **Type quit to exit from restore prompt.**



4. cpio Command

The cpio command is a more sophisticated backup tool than **tar**.
To do a backup, use cpio with a search command, such as find.

The basic structure is: **find -name string -print | cpio -o options > directory**.
The -name option for find lets you search for a string enclosed in double quotation marks. Meta characters can be used. The bar character (|) redirects the output of find to cpio .

The -o flag sets cpio to create an archive file for backing up.

The target is a directory.

The > redirection operator redirects files to the location for the back up. Typically, this location is on a removable device.

5. cpio restore:

To extract files from a cpio backup use the -i (input) mode:

cpio -i options < device files >

6. Tar Command

Tar Command is used to compress and archive and to extract files.

- **To create archive**

  Tar –cvf newfile.tar files – where files represent the files to be copied and newfile.tar is the new file name of the tar.

- **To View archive**

  Tar –tvf newfile.tar – This will list the contents of newfile.tar file.

- **To Extract archive**

  Tar –xvf newfile.tar – This will extract the contents of newfile.tar in a folder named newfile in the current directory.

7. Scheduling backup using crontab utility

8. Package Management

Red Hat's Package Manager (RPM)

RPM is a powerful software manager. It can install, remove, query, and verify the software on your system. Rpm is more than a Red Hat specific tool. Many other modern distributions, such as Caldera and SuSe, use rpm too.

The package info is split into three pieces. The first piece is the package name. The second is the software version number. And, the third is the package build number. All three are separated by dashes. The convention for rpm files is to have the architecture preceding the .rpm extension. Some packages, like man pages, will have *noarch* in the file name. It means that the package is not dependant on the kind of CPU you have.

1. Querying Your System

The first thing you should do is look and see what software you have installed on your system. Here is the command to use:

rpm -qa | more

or

rpm –qa httpd

The *-q* tells rpm you want the query operation.
The *-a* option tells rpm you want to list all the packages.

We can query individual packages like this:

rpm -qi httpd

The *i* query option requires a package name. Notice that *httpd* is used and not *httpd-2.0.52 -22.ent.rpm*

Rpm is smart enough to use the package name without the version info. Supplying the version info will cause an error (but will not harm your system).

To see which files it installed on your system by the *httpd* package

```
rpm -ql httpd
```

2. Installing New Software

Lets look at the command to add new software:

```
rpm -ivh httpd-2.0.52 -22.ent.rpm
```

The *-i* is the install switch.
The *-v* for verbose messages in case the installation fails.
The *-h* option shows our progress with hash marks.

A variation on an install is an **upgrade**. An upgrade is used when you want to put a more recent package in place of something that is currently installed. The upgrade syntax is exactly the same as an install, but you replace the *-i* with a *-U*. (Notice it is a capital U) If a new version of *httpd* comes out, rpm will take care of removing all the old pieces when you upgrade.

```
rpm -Uvh httpd-2.0.62 -22.ent.rpm
```

One last thing that I should mention is that we are installing binary packages. My Intel chip is not binary compatible with an Alpha and so on. The convention for rpm files is to have the architecture preceding the .rpm extension. Some packages, like man pages, will have *noarch* in the file name. It means that the package is not dependant on the kind of CPU you have.

3. Removing Unwanted Software

A major advantage to a packaging system like rpm is its ease to erase so ftware. Here is how you do it:

```
rpm -e httpd
```

Occasionally there may be an error that the package cannot be removed because other software depends on it. We can avoid the dependency check with the *--nodeps* option.

```
rpm -e --nodeps httpd
```

4. Verifying Installed Packages

Verifying a package compares information about the installed files in the package with information about the files taken from the original package and stored in the rpm

database. Among other things, verifying compares the size, MD5 sum, permissions, type, owner, and group of each file. Only the discrepancies are displayed.

Rpm can verify all of your files to see if you were left backdoors or other surprises. Here is how:

rpm –Va

This command will verify all of the files on your system. The syntax should remind you of how you queried your software. Some of the files it reports will be normal. For example, almost everyone will add a nameserver into /etc/resolv.conf. The file has changed since it was originally installed, but it is not a bad change.

5. Advanced Queries

What if you find a file and have no idea what it is or where it came from? Rpm can query that file and show you the package it originated from like this:

rpm -qf /usr/bin/uptime

This command is a little different because it requires the full pathname. Rpm cannot follow symbolic links to a file.

6. Common Errors

Sometimes a package is not removed cleanly. Here is the situation, you try to install something and rpm says its already installed. You then try to remove it, and rpm says that is *not* installed. What can you do?

rpm -ivh --force package-1.0-5.i386.rpm

The *--force* option is our solution.

7. RPM Command Examples

Command	Effect
rpm -qa | less	list all installed software packages
rpm -q httpd	show the version of the httpd package, if it is installed
rpm -qa | grep httpd	show all installed packages that have httpd in their name
rpm -ql httpd	list all files in the httpd package
rpm -qd httpd	list all documentation files in the httpd package
rpm -qc httpd	list all configuration files in the httpd package
rpm -qi httpd	display information about the package

rpm -V httpd	verify that the httpd package is correctly installed
rpm -qf /etc/passwd	determine which package the /etc/passwd file

8. Signing Built RPMs

Signing RPMs adds an extra level of trustworthiness to your RPMs. A digital signature helps establish that the package comes from you, really you, and not from someone masquerading as you. Unfortunately, the RPM system requires a bit of set up work before you can sign RPMs.

9. Checking that the GPG software is installed

To sign packages, you need to ensure that you have the gpg command installed and configured. To check that this command is installed, use a command like the following:

$ rpm -qf `which gpg`
gnupg-1.0.7-6

This shows that the command is available.

PGP stands for Pretty Good Privacy

GPG stands for GNU Privacy Guard, a free, open-source implementation of PGP from the GNU project

10. Verifying packages using public key

Public key location - /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

11. Importing the key

#rpm –import RPM-GPG-KEY-redhat-release

Yellowdog Updater Modified [YUM]

RedHat and other RPM based Linux distributions contain one drawback that it is very hard to upgrade or install software's because of dependency issues. **Yum** stands for **Y**ellowdog **U**pdater, **M**odified for Redhat based distributions. Yum automatically computes dependencies and figures out what steps need to occur in order to install packages. It makes it much easier to maintain groups of machines without having to manually update each one using rpm.

1. **Configuration file**

   /etc/yum.conf

2. **Repository Location**

   /etc/yum.repos.d/<filename>.repo

3. **To install**

   yum install abc
   yum install abc*

4. **To Remove**

   yum remove abc
   yum remove abc*

Yum resolves dependency by using its repository. Creating a repository is important for using yum.

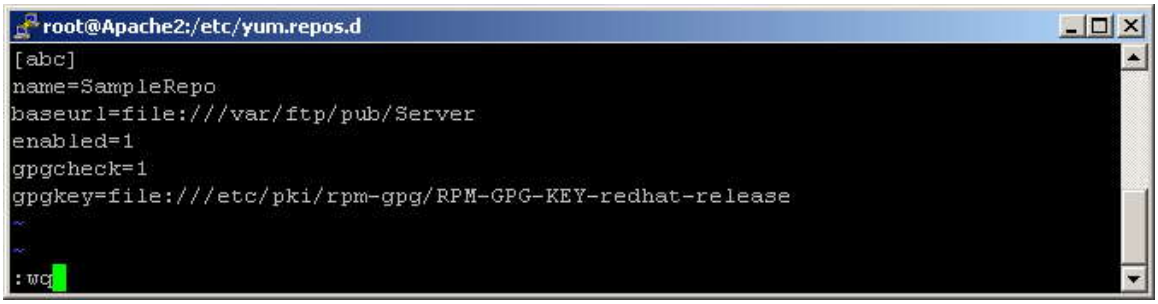
5. **Creating a local repository**

   1.) Copy all the rpm files on the cd to a location such as /var/ftp/pub/Server

   2.) rpm –ivh createrepo*

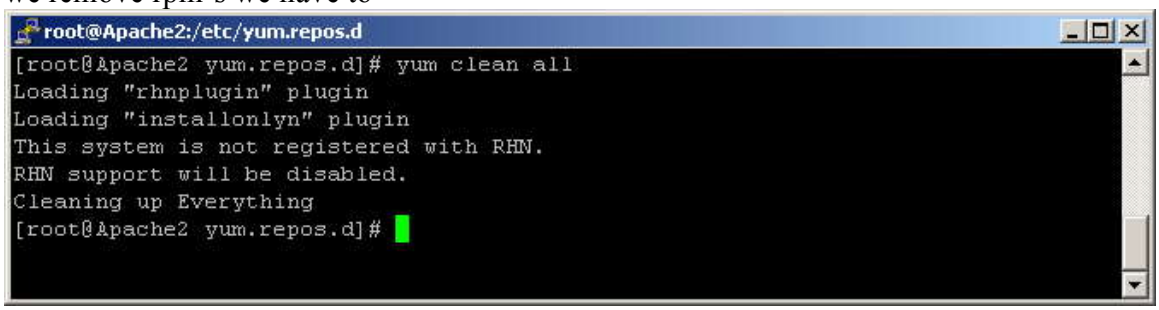
   3.) createrepo –v /var/ftp/pub/Server

   4.) vi /etc/yum.repos.d/abc.repo



The base url on other clients can be http or ftp if they intend to access this directory and the respective service is enabled for this folder

Yum maintains a repository count in the cache, so if we add more rpm's to the repository or if we remove rpm's we have to



6. Other YUM commands

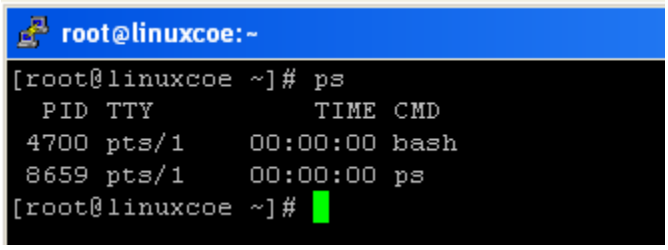
yum list all
yum list available
yum list installed
yum list extras
yum update <package>

9. Process Management

1. Listing and managing all processes running in the system (ps, pgrep):

ps:

ps gives a snapshot of the current processes. If you want a repetitive update of this status, use top.

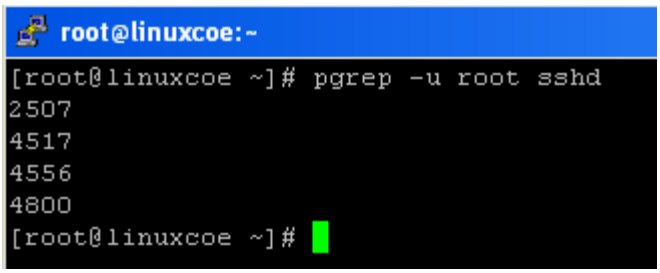


Switch	Description
-A	select all processes
-N	negate selection
-a	select all with a tty except session leaders
-d	select all, but omit session leaders
-e	select all processes
T	select all processes on this terminal
a	select all processes on a terminal, including those of other users
g	really all, even group leaders (does nothing w/o SunOS settings)
r	restrict output to running processes
x	select processes without controlling ttys

--deselect negate selection

Pgrep:

pgrep looks through the currently running processes and lists the process IDs wh ich matches the selection criteria to stdout. All the criteria have to match



1.6 *EXAMPLES*

Example 1: Find the process ID of the **named** daemon:

unix$ pgrep -u root named

Example 2: Give detailed information on all **xterm** processes:

unix$ ps -fp $(pgrep -d, -x xterm)

2. killing the processes:

kill:

To kill a process, you need to know its process ID (PID). The easiest way to check is by ps with grep, let say I wanna check the PID of gaim.

ps aux | grep gaim

Let say it returns PID with numbers 5678, Then now you can kill and check with ps and grep again to confirm whether the process have successfully kill.

kill 5678

Alternatively, you can kill processes by specified process name using pkill, again I wanna kill gaim.

pkill gaim

Kill a process?
You can choose a signal to send to the process, checks the kill manual for available signals to send.

man kill

Usually if you process was unable to kill, first check whether you need to have root privilege to kill? certain process is instantiate by root, to kill you must have the same level of privilege.

sudo kill 5678

Second, try to kill with -1 and -2 before uses -9.

kill -1 5678

Okay, the process is really irritating, let me force kill it,

kill -9 5678

Some process may have multiple instances, killing the process one by one with PID is nightmare, so kill them all one shot!

killall gaim

Still there? grrrrrrrrrr!

sudo killall -9 gaim

PKILL:

pkill will send the specified signal (by default **SIGTERM**) to each process instead of listing them on stdout.

Example : Make **syslog** reread its configuration file:

unix$ pkill -HUP syslogd

3. Pushing Jobs to background and pulling jobs from background (bg, fg, jobs)

Running jobs in background

running jobs in background providing & followed by a command will make the command to run in background.

Sleep command runs in background:

[root@penrose ~]# sleep 100 &

```
[1] 21594
[root@penrose ~]#
```

Jobs command lists all the running jobs:

```
[root@penrose ~]# jobs
[1]+  Running                 sleep 100 &
[root@penrose ~]#
```

For example, you have started a job without &, so you will not get prompt till the job is completed, but you have few more jobs to be completed in the mean time. Therefore, you would like to push this running job to background in order to access the prompt.

What will you do?

You can use bg command to push the job to background, and fg command to pull the job running in bckground to foreground.

Sleep command runs in foreground:

```
[root@penrose ~]# sleep 100
```

Here, press Ctrl+Z to suspend the job

```
[1]+  Stopped                 sleep 100
[root@penrose ~]#
```

Ctrl+Z will suspend the job temporaryly.

Jobs command display that sleep command is running in foreground:

```
[root@penrose ~]# jobs
[1]+  Stopped                 sleep 100
[root@penrose ~]#
```

bg command pushes the job to background:

```
[root@penrose ~]# bg 1
[1]+ sleep 100 &
[root@penrose ~]#
```

Syntax is #bg job_number

Jobs command display that sleep command is running in background now:

```
[root@penrose ~]# jobs
[1]+  Running                 sleep 100 &
[root@penrose ~]#
```

fg command brings the sleep command to foreground now:

```
[root@penrose ~]# fg 1
sleep 100
```

Syntax is #fg Job_number

Pressing Ctrl+Z will again suspend the job.

4. /proc file system and it's importance:

/proc is often referred to as a *virtual file system*

These virtual files have unique qualities. Most of them are listed as zero bytes in size and yet when one is viewed, it can contain a large amount of information. In addition, most of the time and date settings on virtual files reflect the current time and date, indicative of the fact they are constantly updated.

```
[root@wiki ~]# cat /proc/pci          ------lists plugged in PCI devices

[root@wiki ~]# cat /proc/ioports      ----- list what I/O are used

[root@wiki ~]# cat /proc/meminfo      -----info about memory usage

[root@wiki ~]# cat /proc/scsi/scsi    ----- lists scsi devices

[root@wiki ~]# cat /proc/modules      ----- lists all loaded kernel modules
```

To get CPU information, Type the following command:

```
[root@wiki ~]# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 15
model           : 4
model name      : Intel(R) Pentium(R) 4 CPU 3.00GHz
stepping        : 3
cpu MHz         : 2992.935
cache size      : 2048 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 1
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 3
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm pni monitor ds_c
pl est cid xtpr
bogomips        : 5989.50

processor       : 1
vendor_id       : GenuineIntel
cpu family      : 15
model           : 4
model name      : Intel(R) Pentium(R) 4 CPU 3.00GHz
stepping        : 3
cpu MHz         : 2992.935
cache size      : 2048 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 1
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
```

```
cpuid level     : 3
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm pni monitor ds_c
pl est cid xtpr
bogomips        : 5985.17

[root@wiki ~]#
```

To get all block and character devices, run the following command:

```
[root@wiki ~]# cat /proc/devices
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  6 lp
  7 vcs
 10 misc
 13 input
 14 sound
 29 fb
 36 netlink
 89 i2c
116 alsa
128 ptm
136 pts
162 raw
180 usb
226 drm

Block devices:
  1 ramdisk
  2 fd
  3 ide0
  7 loop
  8 sd
  9 md
 65 sd
 66 sd
 67 sd
 68 sd
```

```
 69 sd
 70 sd
 71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
253 device-mapper
254 mdp
[root@wiki ~]#
```

In the same way, you run the following commads:

```
#cat /proc/filesystems   Lists supported filesystems
# cat /proc/kmsg         As root !  kernel messages . . close with Ctrl+C
# cat /proc/modules      A list of loaded drivers
# cat /proc/partitions   All partitions your kernel recognizes
#cat /proc/pci           devices in the pci slots
# cat /proc/version      Kernel version, the gcc it was commpiled with, etc . . .
```

10. Performance Tuning

1. About the commands vmstat, iostat, free, top, etc..

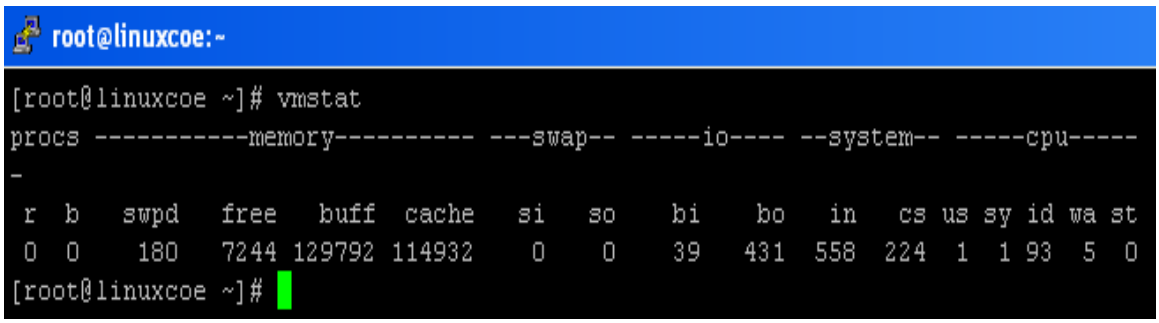
vmstat: reports information about processes, memory, paging ,block IO, traps, and cpu activity.

Options

The *-n* switch causes the header to be displayed only once rather than periodically.

delay is the delay between updates in seconds. If no delay is specified, only one report is printed with the average values since boot.

count is the number of updates. If no count is specified and delay is defined, *count* defaults to infinity.



Iostat:

iostat is a command line I/O performance monitoring utility. Displays an overview of CPU utilization, along with I/O statistics for one or more disk drives.

Free:

The free command displays system memory utilization. Here is an example of its output

```
root@linuxcoe:~
[root@linuxcoe ~]# free
             total       used       free     shared    buffers     cached
Mem:        375336     368464       6872          0     129840     114964
-/+ buffers/cache:     123660     251676
Swap:      2096440        180    2096260
[root@linuxcoe ~]#
```

to display memory utilization every two seconds (the default display interval), use this command:

[root@linuxcoe ~]# watch free

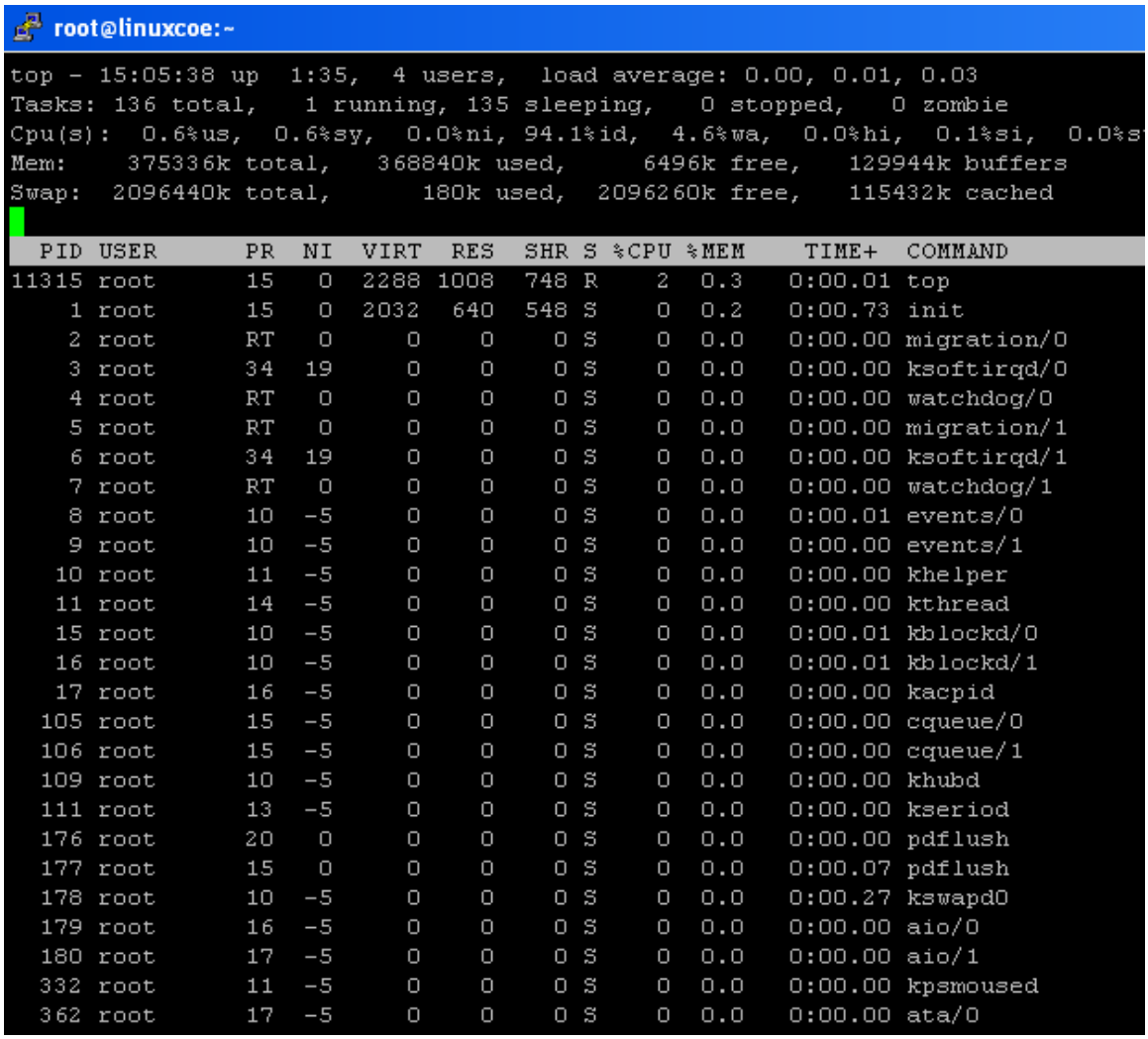
```
root@linuxcoe:~
Every 2.0s: free

             total       used       free     shared    buffers     cached
Mem:        375336     368460       6876          0     129888     115340
-/+ buffers/cache:     123232     252104
Swap:      2096440        180    2096260
```

watch command issues the free command every two seconds, after first clearing the screen. This makes it much easier to see how memory utilization changes over time, as it is not necessary to scan continually scrolling output. You can control the delay between updates by using the -n option, and can cause any changes between updates to be highlighted by using the -d option, as in the following command:
watch -n 1 -d free

```
root@linuxcoe:~
Every 1.0s: free

             total       used       free     shared    buffers     cached
Mem:        375336     368584       6752          0     129928     115348
-/+ buffers/cache:     123308     252028
Swap:      2096440        180    2096260
```

Top

Top command displays CPU utilization, process statistics, and memory utilization.



Kernel running parameters

Today we are going to see about Linux kernel.

You may all know that kernel is a core of Linux operating system. The entire kernel parameters can be found under /proc/sys file system. You can also view the kernel parameters by running the following command:

#/sbin/sysctl –a --> This displays current kernel parameters

[root@wiki ~]# /sbin/sysctl -a |more
sunrpc.tcp_slot_table_entries = 16
sunrpc.udp_slot_table_entries = 16
sunrpc.max_resvport = 1023

```
sunrpc.min_resvport = 650
sunrpc.nlm_debug = 0
sunrpc.nfsd_debug = 0
sunrpc.nfs_debug = 0
sunrpc.rpc_debug = 0
dev.scsi.logging_level = 0
dev.raid.speed_limit_max = 200000
dev.raid.speed_limit_min = 1000
dev.cdrom.check_media = 0
dev.cdrom.lock = 1
dev.cdrom.debug = 0
dev.cdrom.autoeject = 0
dev.cdrom.autoclose = 1
dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17
dev.cdrom.info =
dev.cdrom.info = drive name:          hda
dev.cdrom.info = drive speed:         48
dev.cdrom.info = drive # of slots:    1
dev.cdrom.info = Can close tray:
--More--
```

If you make changes in some of the system configuration, it will not be taken into effect till the next reboot. However, if you are making change in the kernel parameter directly, it will be effective immediately.

for example, change the hostname, or changing the Nis Domain Name, or locking the CD Drive etc... will be taken into effect only after rebooting the system. If you make these changes in kernel, these will be effective immediately.

For example:

#/sbin/sysctl -w kernel.hostname="test1.linuxcoe.com" -- Changes the hostname online

```
[root@penrose ~]# hostname
penrose.linux.com
[root@penrose ~]# /sbin/sysctl -w kernel.hostname="test1.linux.com"
kernel.hostname = test1.linux.com
[root@penrose ~]#
[root@penrose ~]#
[root@penrose ~]# hostname
test1.linux.com
[root@penrose ~]#
```

2. Creating empty file and assigning this to swap and enabling the swap

Adding swap space

Swap partition in Linux provides virtual memory space to run the larger programs. You can create a swap partition, during installation itself. In case, you would like to add space with the existing swap space, How will you do it?

Here are the steps:

To display the swap space enabled on your Linux system, run the command "swapon –s" as below:

```
[root@wiki ~]# swapon -s
Filename                        Type            Size    Used    Priority
/dev/sda3                       partition       1052248 176     -1
[root@wiki ~]#
```

The above output shows '/dev/sda3' is now configured as swap partition.

To add swap space with this, you should create an empty file with the required size using the command "dd" as below:

```
[root@wiki ~]# dd if=/dev/zero of=/swapfile bs=1024 count=65536
65536+0 records in
65536+0 records out
[root@wiki ~]#
```

Configure this empty file as swapspace:

```
 [root@wiki ~]# mkswap /swapfile
Setting up swapspace version 1, size = 67104 kB
[root@wiki ~]#
```

Enable this swap space:

```
[root@wiki ~]# swapon /swapfile
```

Now, "swapon –s" displays the swap space enabled at present:

```
[root@wiki ~]# swapon -s
Filename                        Type            Size    Used    Priority
/dev/sda3                       partition       1052248 176     -1
/swapfile                       file            65528   0       -3
[root@wiki ~]#
```

At any time, you can disable swap space by running the command "swapoff"

3. Disabling swap added

```
[root@wiki ~]# swapoff /swapfile
[root@wiki ~]# swapon -s
Filename                        Type            Size    Used    Priority
/dev/sda3                       partition       1052248 176     -1
[root@wiki ~]#
```

11. X Window Configuration

Linux server can be connected from any windows machine using telnet or ssh service. In this case, anything can be done only via Command line. X server /X client allows to take Linux GUI from remote machine. You can also use VNC to take Linux GUI from remote machine.

Virtual Network Computing (VNC) allows one to view and operate the console of another computer remotely across the network. It is also known generically as RFB or Remote Frame Buffer. This document will cover the use of a VNC client running on Microsoft Windows to view and operate Redhat Linux remotely.

VNC Server :

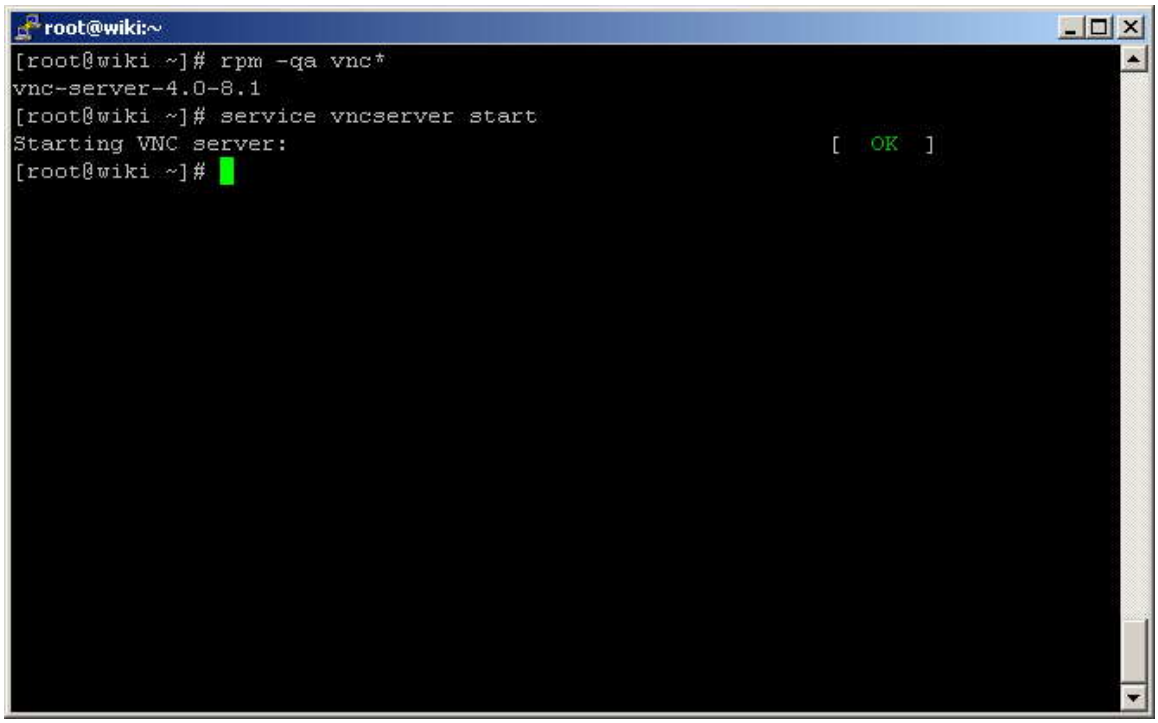
- VNC is a Cross Platform Networking
- A VNC server is the machine you want to access remotely.

VNC Server Configuration on Linux

Login to Remote Linux Server using **PUTTY or SSH** & do these steps :

1. Check whether VNC Server in your system, type the following command
$ rpmquery vnc-server
2. if already installed, following rpm package will be displayed
vnc-server-4.1.1-10
3. If the above package s not installed, download and installing your system.
4. How to install the VNC Server

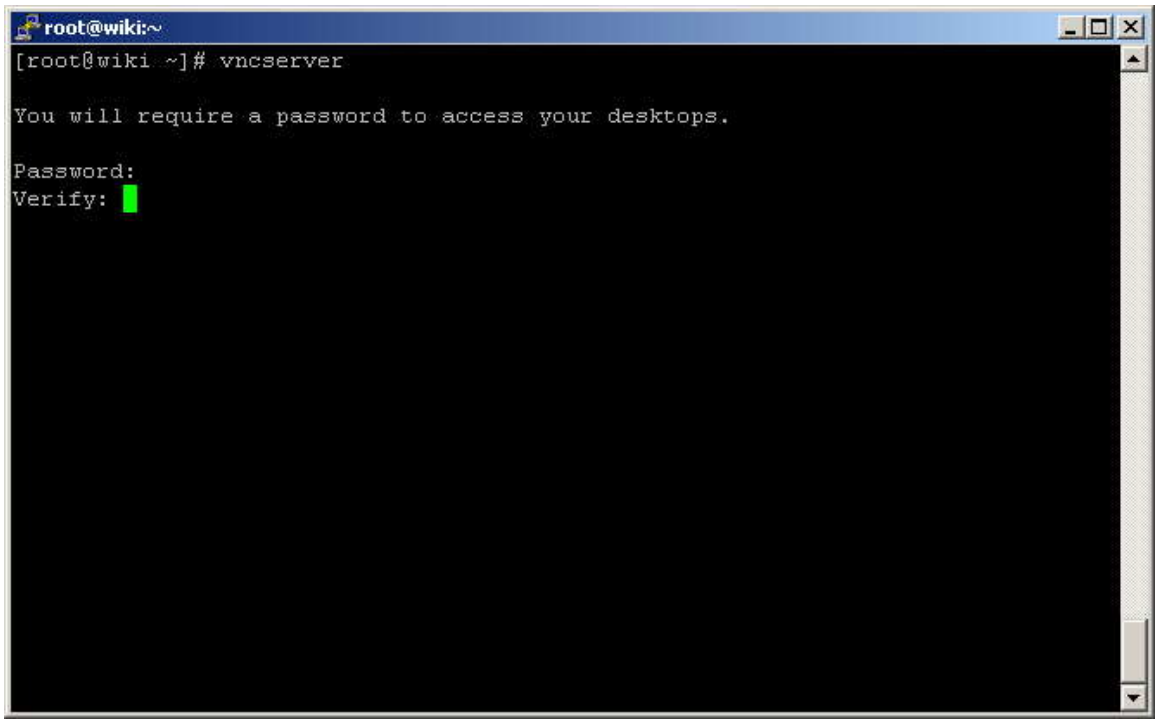
Download the vnc-4.1.1-10.rpm package files.
Run the vnc rpm package, type the following command
rpm -ivh vnc-4.1.1.-10.rpm
Press enter
Successfully installed.



5. After installed the vnc server.
6. type the following command

$ vncserver

Display the following lines
you will require a password to access your desktop
password : *******[any names]
press enter
verify : *******[above the same name]
press enter



This password is required to access the Linux desktops.

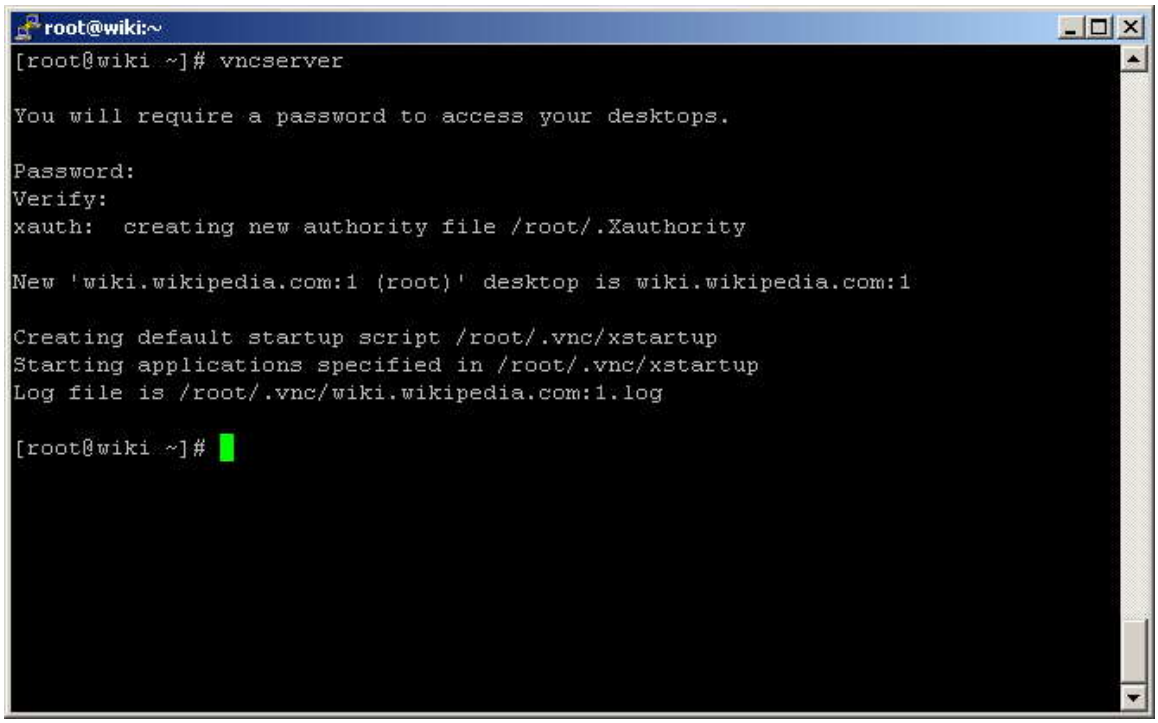
Following few lines will be displayed

New localhost.localdomain:1(user) desktop is localhost.localdomain:1
creating default startup script /home/user/.vnc/xstartup
log file is /home/user/.vnc/localhost.localdomain:1.log

if you will change any password , type the the following command
$ vncpasswd
press enter
display the following lines
password:*******[type any password]
verify:*********[same above password]

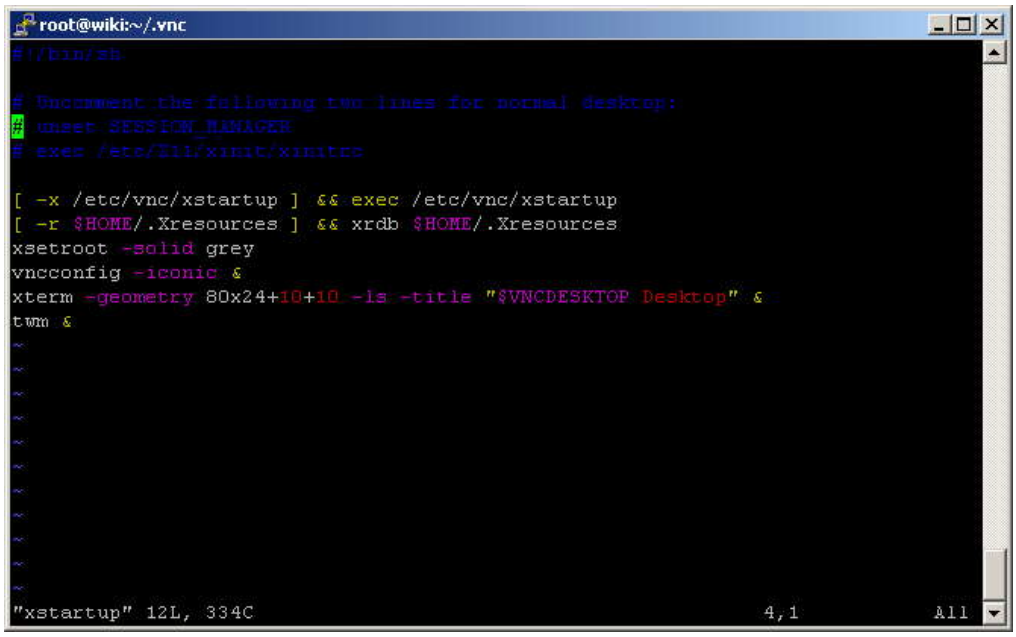
Configuring Your Firewall For VNC :

1. To Open the Security Level Configurations
2. How to Open and configure the Security Level
   start menu
   select system settings
   select security level
   display the security level configurations panel
   Disable the firewall options
3. Click ok and close it.



Customizing the VNC Server :

• open the following file in /home/user/.vnc/xstartup and display the following lins
#!/bin/sh
uncomment the following two lines for normal desktop
unset SESSION_MANAGER
exec /etc/X11/Xinit/Xintrc



In the above file REMOVE the " # " symbol for the below said 2 lines :-

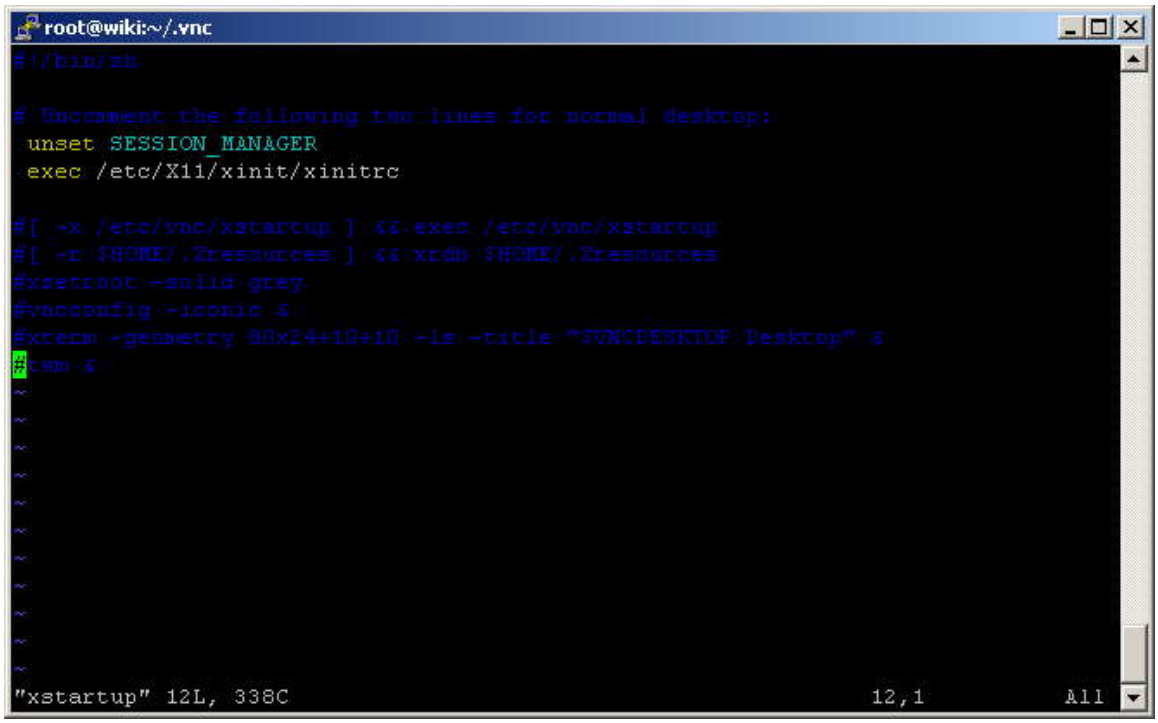
Unset SESSION _MANAGER
Exec /etc/X11/xinit/xinitrc

UNCOMMENT the following lines : -

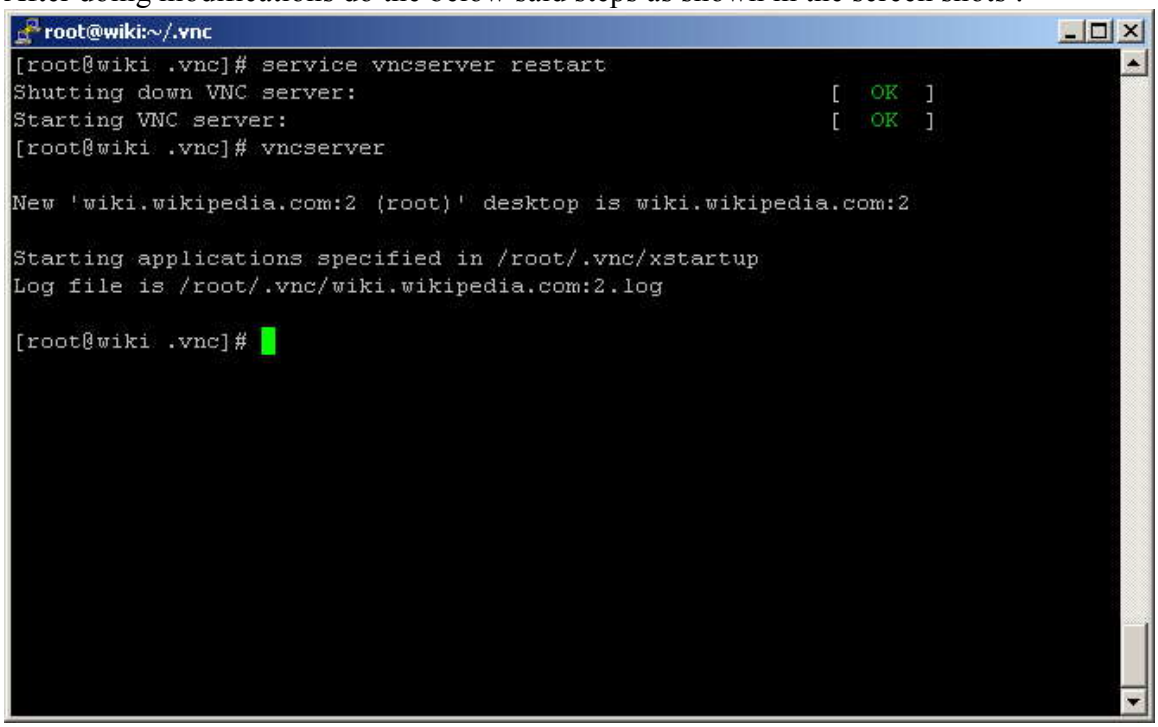
#[-x /etc/vnc/xstartup] && exec /etc/vnc/xstartup
#[-r $HOME/.Xresources] && xrdb $HOME/.Xresources
#xsetroot -solid grey
#vncconfig -iconic &
#xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
#twm &

now you add the following lines should be configured and previosly twolines have commented,
#!/bin/sh
uncomment the following two lines for normal desktop
#unset SESSION_MANAGER
#exec /etc/X11/Xinit/Xintrc
[-x /etc/vnc/xstartup] && exec /etc/vnc/xstartup
[-r $HOME/.Xresources] $$ xrdb $HOME/.Xresources
xsetroot -solid grey
vncconfig -iconic &
xterm -gemotry 80x24+10+10 -ls -title "&VNCDESKTOP Desktop" &

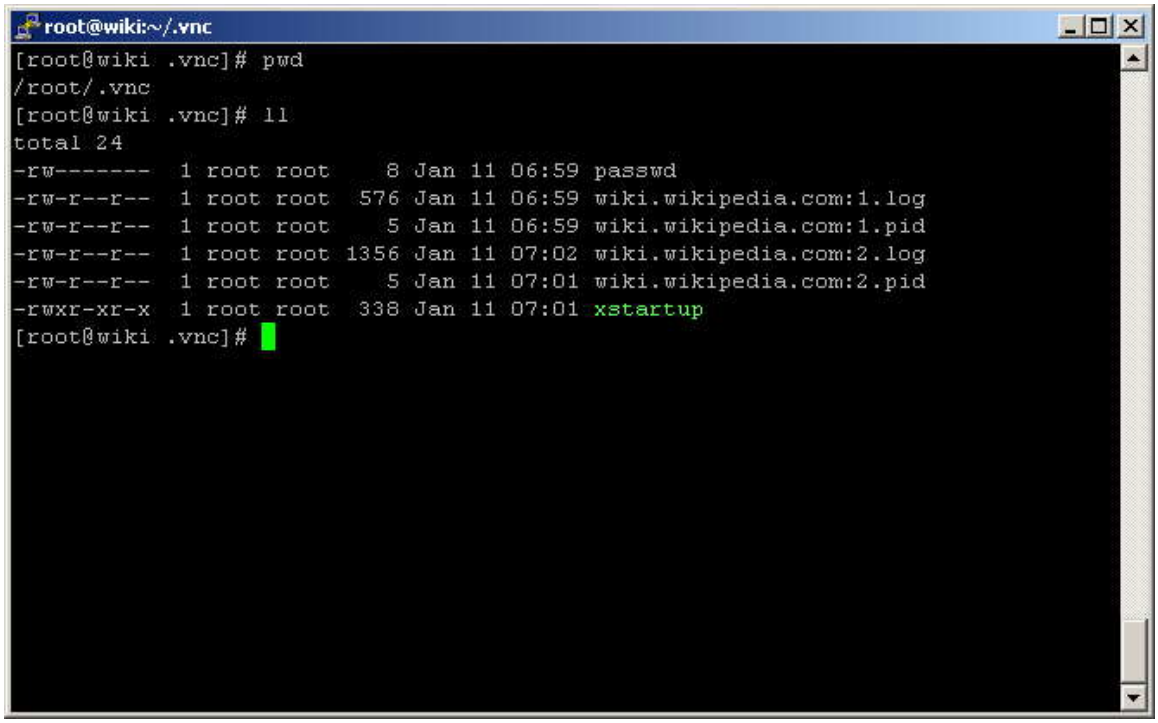
twm &



After doing modifications do the below said steps as shown in the screen shots : -



Path for the file "xtstartup"



Testing the VNC Server:
Step 1: Check Whether VNC Server is running or not.if vncserver is running then start the vnc server service and follow it.

Step 2: Type the command $ vncserver

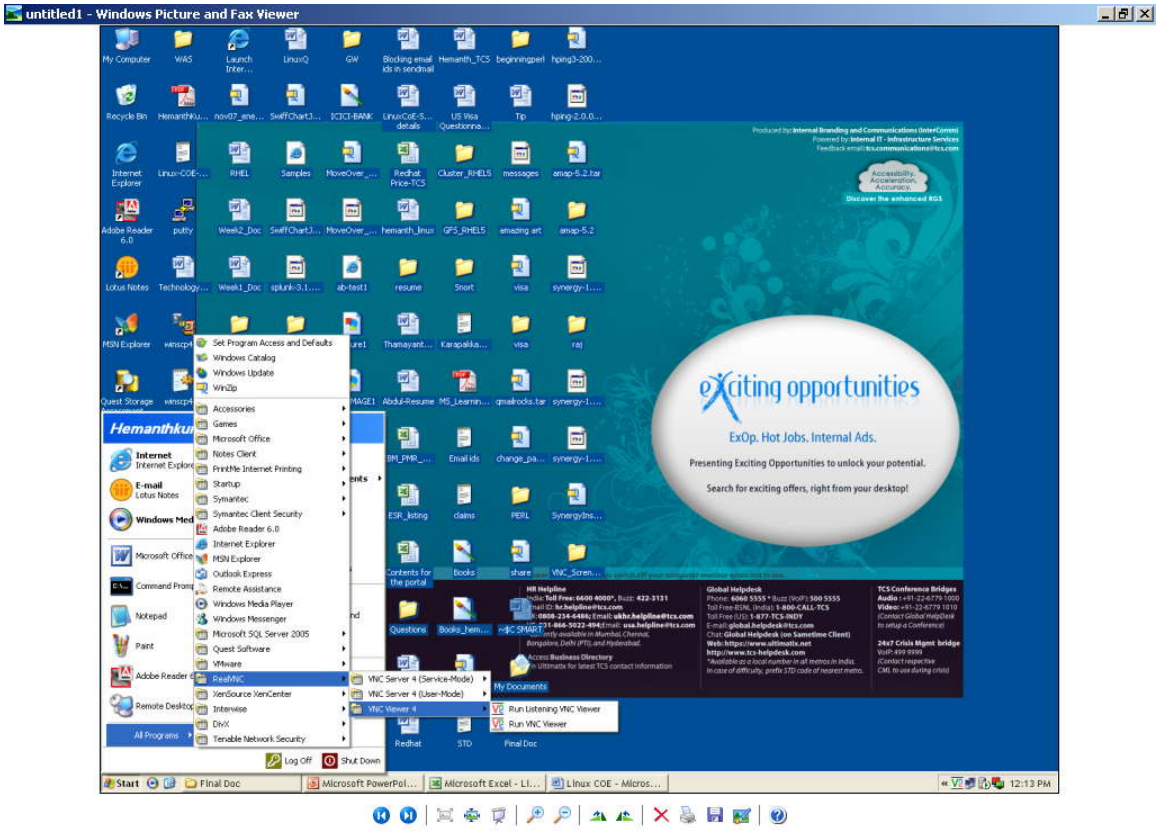
Step 3: Display the following lines.
New localhost.localdomain.com:1(user)
Starting applications specified in /home/user/.vnc/xstartup
log file is /home/user/.vnc/localhost.localdomain.com:1.log

Step 4: Type the command $ vncviewer ipaddress :1

VNC Client configuration in windows:

a.) Download and install the VNC Viewer client sw in the windows desktop.

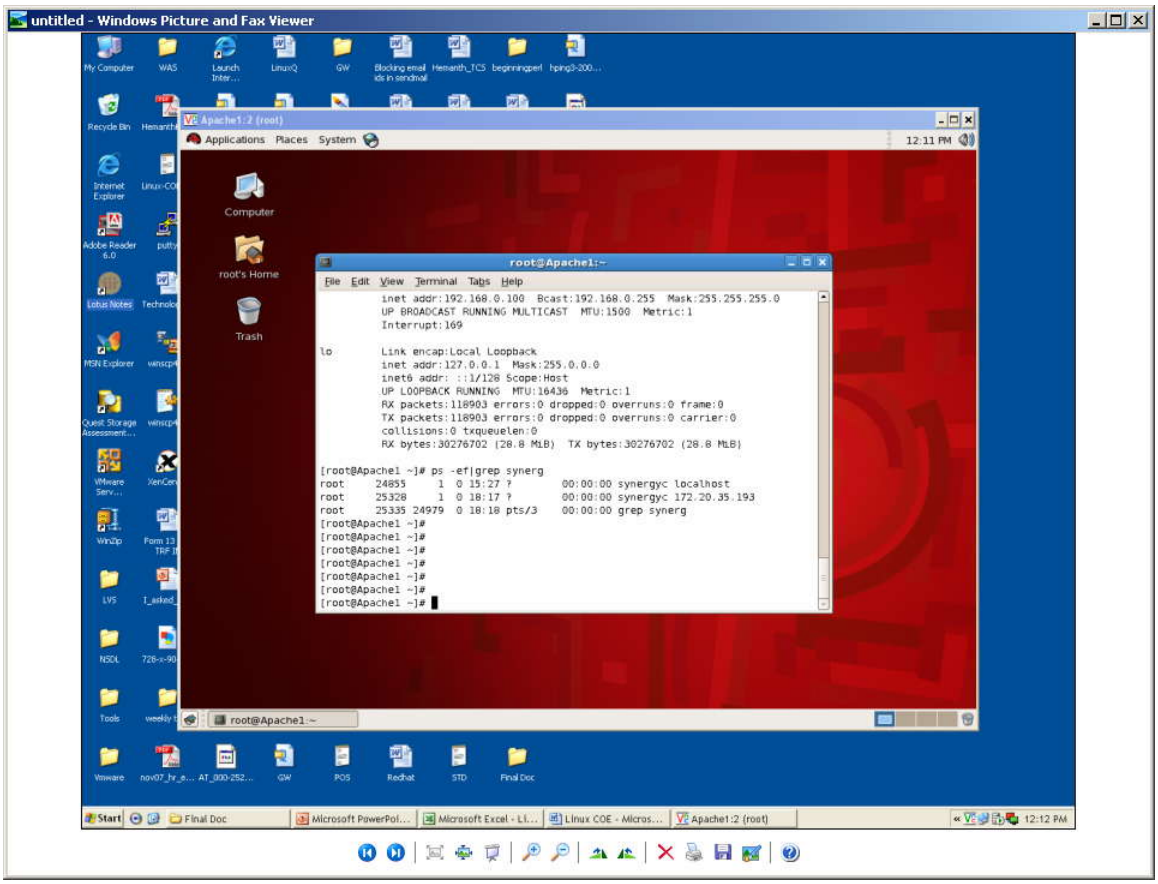
To Start the VNC client follow the path and run the vnc viewer.



Enter the remote linux server ip add and password.

Example:
<ip addres> : 1 (without GUI)
<ip address>:2 (with GUI)

Remote desktop (Redhat Linux) opens thro VNC in windows client.


untitled - Windows Picture and Fax Viewer
Apache1:2 (root)
Applications Places System
12:11 PM
Computer
root's Home
Trash
root@Apache1:~
File Edit View Terminal Tabs Help
inet addr:192.168.0.100 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
Interrupt:169
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:118903 errors:0 dropped:0 overruns:0 frame:0
TX packets:118903 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:30276702 (28.8 MiB) TX bytes:30276702 (28.8 MiB)
[root@Apache1 ~]# ps -ef|grep synerg
root 24855 1 0 15:27 ? 00:00:00 synergyc localhost
root 25328 1 0 18:17 ? 00:00:00 synergyc 172.20.35.193
root 25335 24979 0 18:18 pts/3 00:00:00 grep synerg
[root@Apache1 ~]#
[root@Apache1 ~]#
[root@Apache1 ~]#
[root@Apache1 ~]#
[root@Apache1 ~]#
[root@Apache1 ~]#
[root@Apache1 ~]#
root@Apache1:~
Start
Final Doc
Microsoft PowerPoi...
Microsoft Excel - Li...
Linux COE - Micros...
Apache1:2 (root)
12:12 PM

12. Printing

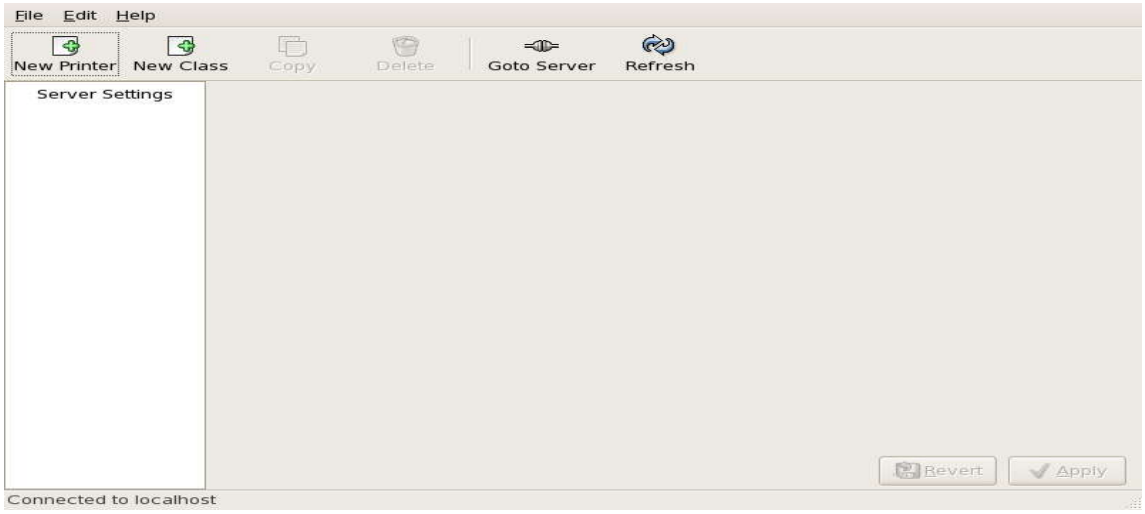
Red Hat Linux includes a graphical utility for configuring local and remote printers without the need to install additional drivers and applications. The **Printer Configuration Tool** uses a step-by-step process that will help to configure a printer faster than editing configuration files manually.

1. Adding a Local Printer

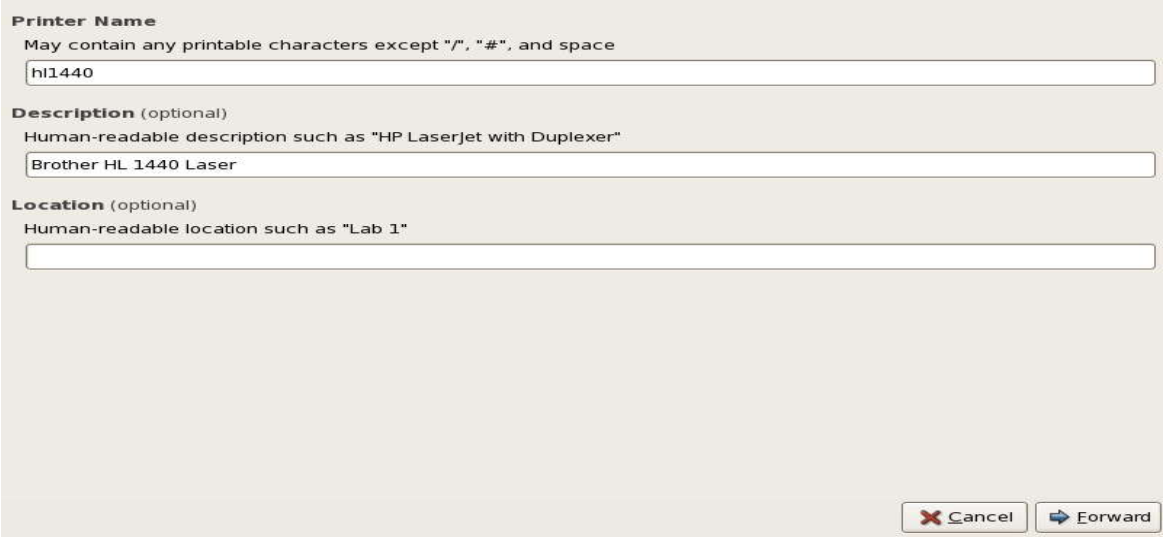
- To add a local printer that is connected to the parallel port or USB port
- Make sure that the CUPS Service (cupsd daemon) is running:

  # service cups restart
- To open the Printer Configuration Window:

  # system-config-printer &

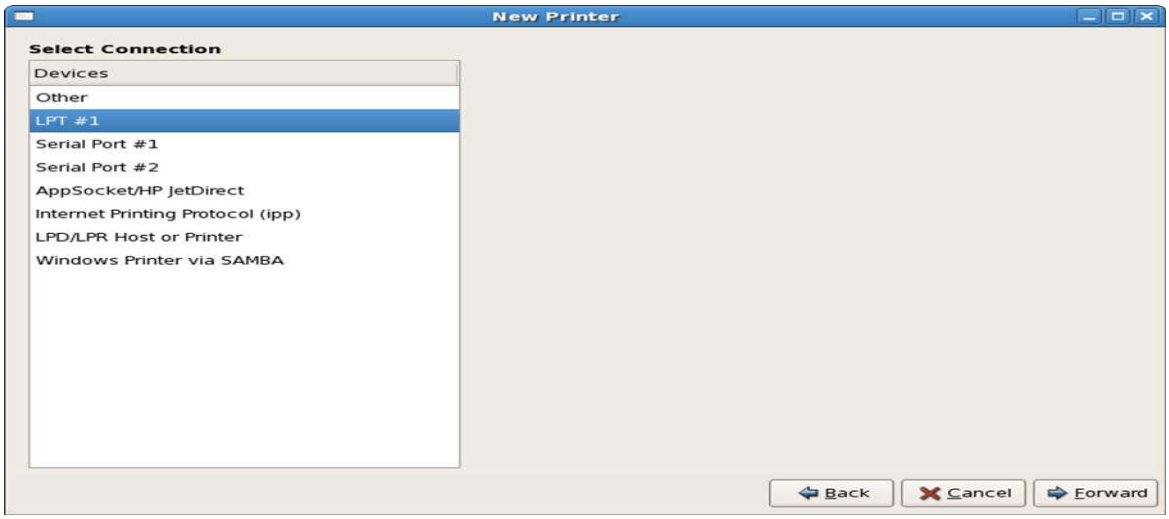


- Click the **New Printer** button in the main **Printer Configuration Tool** window
- Give the Printer Name, Description, Location and Click **Forward** to proceed.



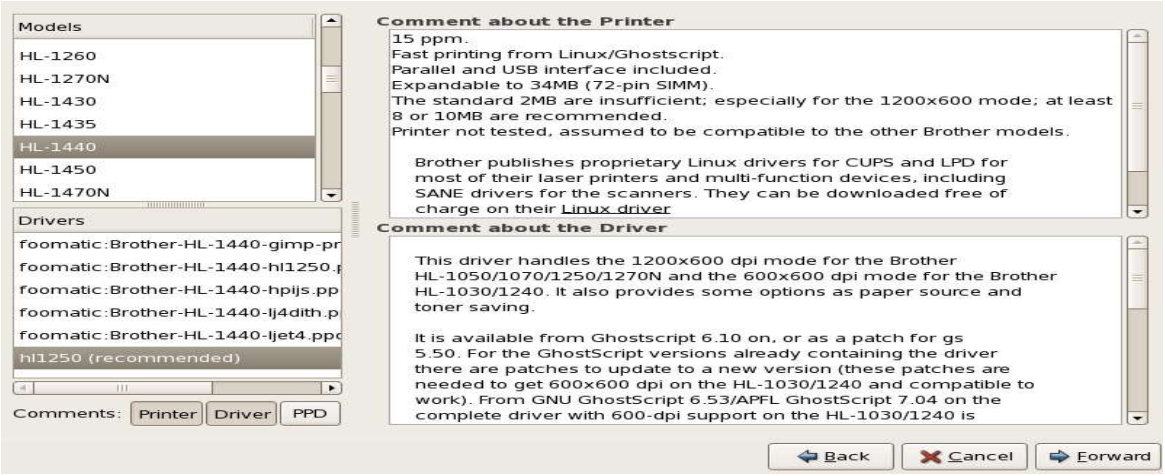
2. Adding a Printer

- If the printer has been automatically detected, the printer model appears in **Select Connection**. Select the printer model and click **Forward** to continue.
- If the printer does not appear automatically, select the device to which the printer is connected (such as **LPT #1** or **Serial Port #1**) in **Select Connection**.



3. Selecting the Printer Model and Finishing

- Select the printer queue type
- Select a Printer from database - Choose the make of the printer from the list of **Makes**. If the printer make is not listed, choose **Generic**.
- Provide PPD file - A PostScript Printer Description (PPD) file may also be provided with the printer. This file is normally provided by the manufacturer. If provided with a PPD file, choose this option and use the browser bar below the option description to select the PPD file and Click **Forward**



4. Confirming Printer Configuration

- The last step is to confirm the printer configuration.
- Click **Apply** to add the print queue if the settings are correct.

5. Printing a Test Page

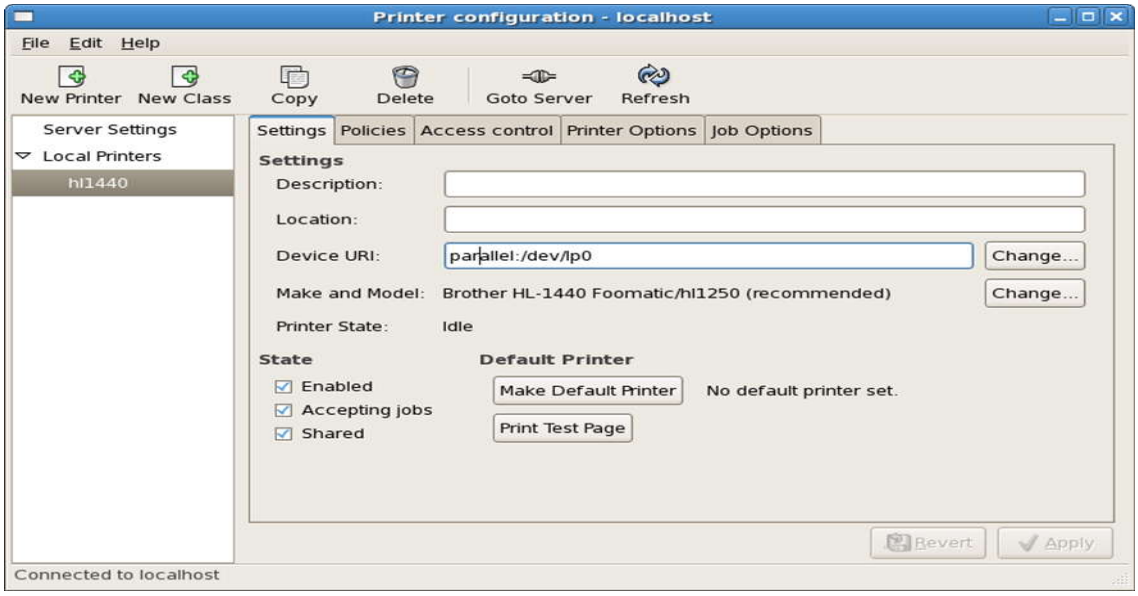
- After Configuring the Printer, to print a test page:
- Select the printer from the printer list, Click **Print Test page** from the Settings tab.

6. Modifying Existing Printers

- To delete an existing printer, select the printer and click the **Delete** button on the toolbar.
- The printer is removed from the printer list after confirming deletion of the printer configuration.

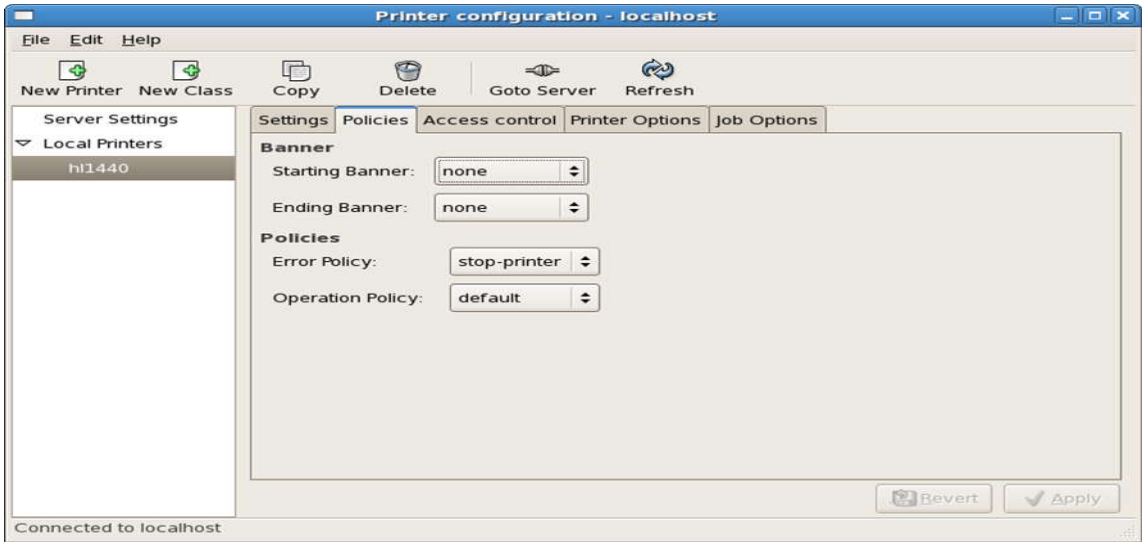
The Settings Tab

- To change printer driver configuration, click corresponding name in the **Printer** list and click the **Settings** tab.
- To modify printer settings such as make and model, make a printer the default, print a test page, change the device location (URI), and more.



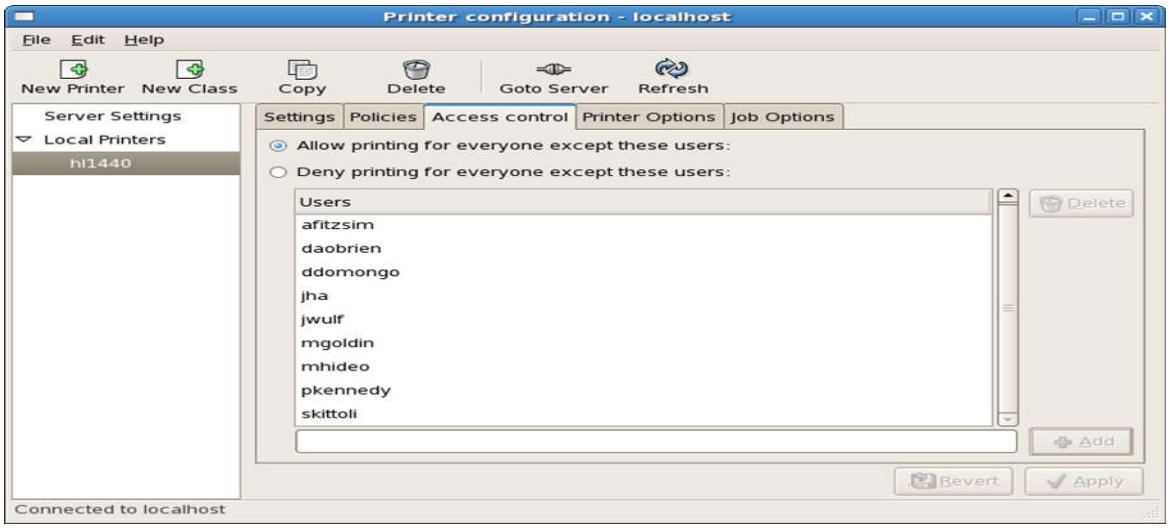
The Policies Tab

- To change settings in print output, click the **Policies** tab.
- Click the **Starting Banner** or **Ending Banner** drop-menu and choose the option that best describes the nature of the print jobs (such as **topsecret**, **classified**, or **confidential**).
- **Error Policy** of the printer can be configured, by choosing an option from the drop-down menu to abort the print job, retry, or stop it.



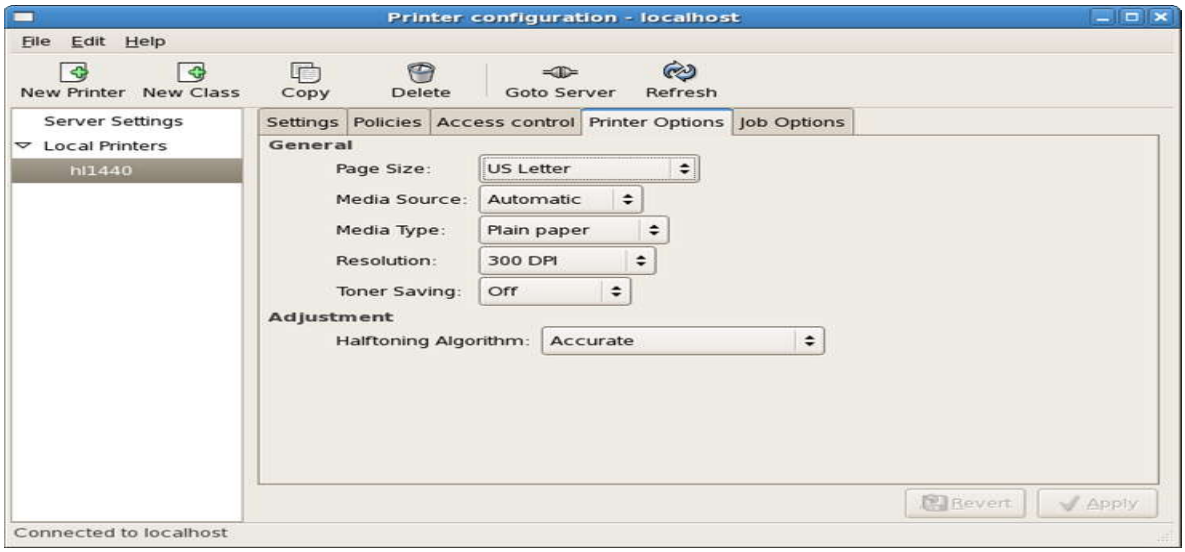
The Access Control Tab

- Change user-level access to the configured printer by clicking the **Access Control** tab.
- Add users using the text box and click the **Add** button beside it. Then choose to only allow use of the printer to that subset of users or deny use to those users.



The Printer and Job Options Tab

The **Printer Options** tab contains various configuration options for the printer media and output.



Printer Options Tab

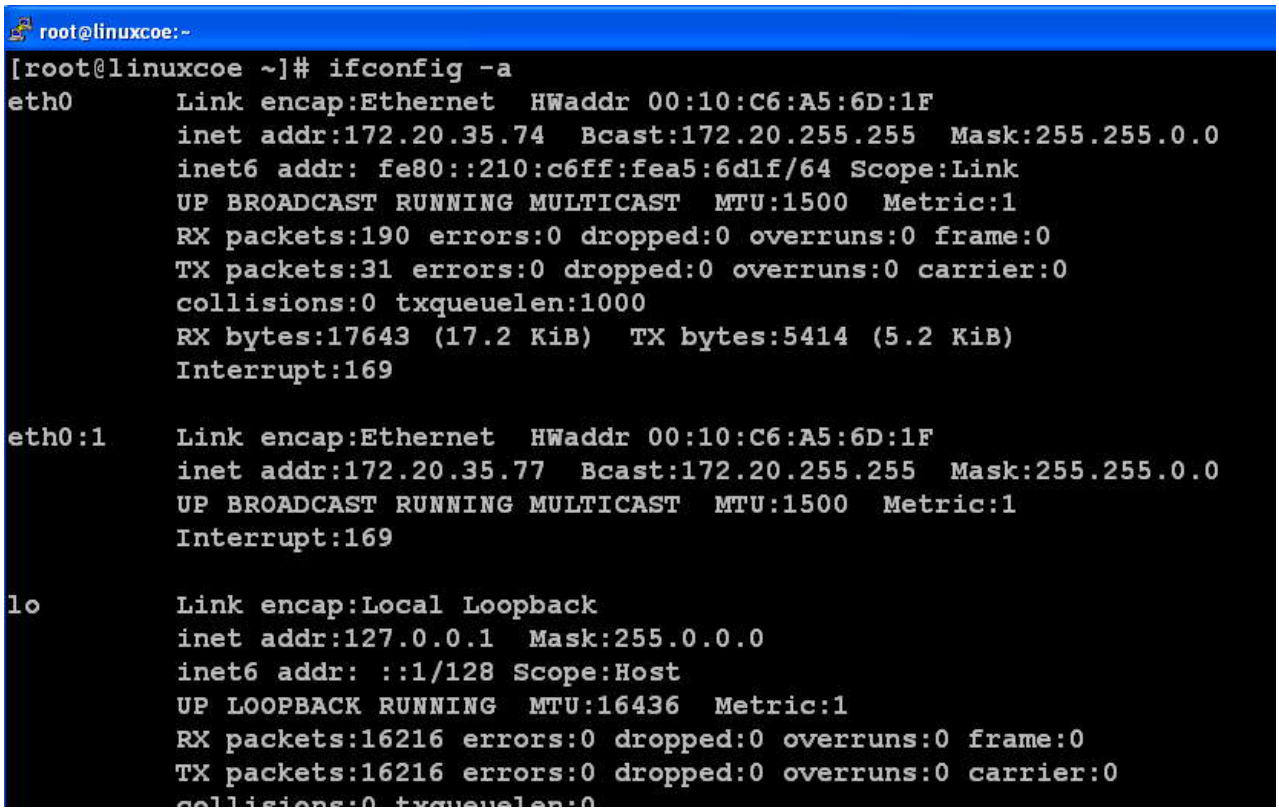
- **Page Size** — Allows the paper size to be selected. The options include US Letter, US Legal, A3, and A4
- **Media Source** — set to **Automatic** by default. Change this option to use paper from a different tray.
- **Media Type** — Allows you to change paper type. Options include: Plain, thick, bond, and transparency.
- **Resolution** — Configure the quality and detail of the printout (default is 300 dots per inch (dpi).
- **Toner Saving** — Choose whether the printer uses less toner to conserve resources.

13. Network Configuration

1. To view the two ipaddress

ifconfig –a

Likewise, we can set multiple IP addresses to a single Ethernet card.



2. Network Configuration to Ethernet Card

We can assign more than one IP address to the linux system although we have one Ethernet card in the system.

3. Configuration file

#vi /etc/sysconfig/network-scripts/ifcfg-eth0

If we open the file, we will get the following

```
root@linuxcoe:~
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
IPADDR=172.20.35.74
NETMASK=255.255.0.0
GATEWAY=172.20.255.254
TYPE=Ethernet
~
```

Here we can edit IPADDR to change IP address what we need. To assign more than one IP address, open a new file as follows

```
root@linuxcoe:~
[root@linuxcoe ~]# cat > /etc/sysconfig/network-scripts/ifcfg-eth0:1
```

Type the lines here as above and set a new IP address in IPADDR, then save the file.

4. To restart the network service

```
root@linuxcoe:~
[root@linuxcoe ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Disabling IPv4 packet forwarding:                          [  OK  ]
Setting network parameters:                                [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
[root@linuxcoe ~]#
```

14. NFS (Network File Service)

What is NFS?

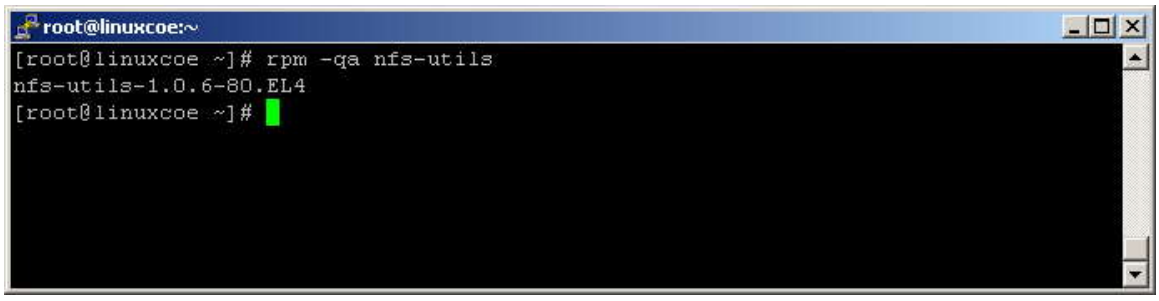
Network **F**ile **S**ystem, a client/server application designed by Sun Microsystems that allows all network users to access shared files.

1. Daemons

portmap – maps calls made from other machines to the correct RPC service.
nfsd – translates NFS requests into requests on local file system.
rpc.mountd – mounts and unmounts file systems.

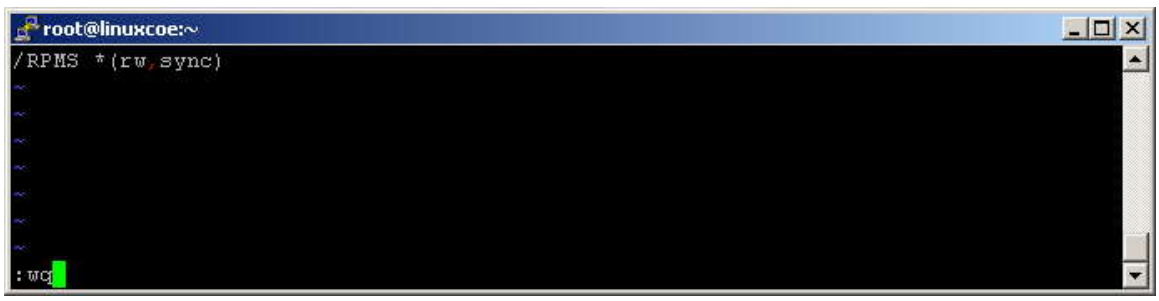
2. Packages

nfs-utils-1.0.6-65.EL4.i386.rpm



3. Configuration file

vi /etc/exports

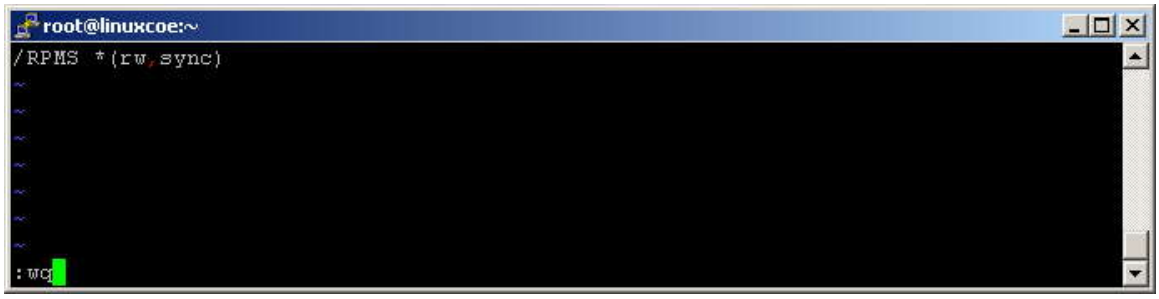


4. Installing NFS

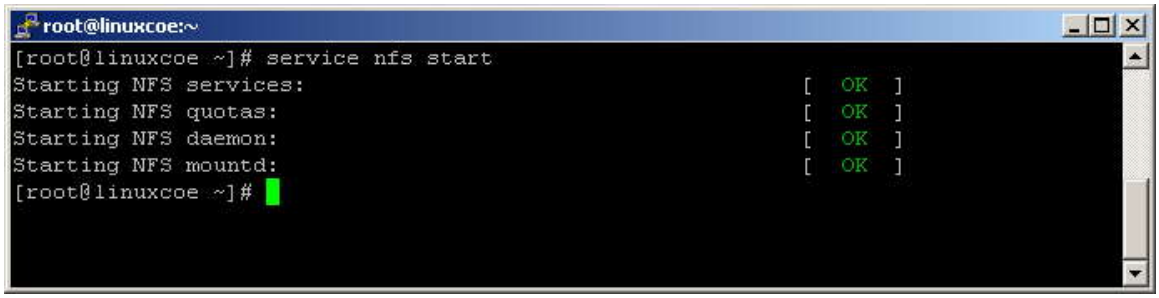
#rpm –ivh nfs-utils-1.0.6-65.EL4.i386.rpm

5. Configuring NFS

- **Add the directory (to be shared) in /etc/exports file**



- **2. Starting the NFS Service**



.

- **Exporting the nfs share**

  # exportfs – a

- **To See the exported directory**

  # showmount -e

6. Accessing NFS Share from the client

Login to the client machine (in this case it's the wiki server ip -172.20.35.99) and create a directory that can be used for mounting the nfs share and mount it as follows

```
root@wiki:/
[root@wiki /]# mkdir newmount
[root@wiki /]# mount 172.20.35.74:/RPMS /newmount
[root@wiki /]#
```

7. To check if it is mounted ….

```
root@wiki:/
[root@wiki /]# mount
/dev/sda2 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sda1 on /boot type ext3 (rw)
/dev/md0 on /data type ext3 (rw)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
/dev/sda7 on /new type ext3 (rw)
/FC6/FC-6-i386-disc2.iso on /mnt type iso9660 (rw,loop=/dev/loop0)
172.20.35.74:/RPMS on /newmount type nfs (rw,addr=172.20.35.74)
[root@wiki /]#
```

Check the last line… It shows that the /RPMS folder on 172.20.35.74 (penrose) server has been mounted on /newmount that we created on wiki (172.20.35.99) server.

To make this change permanent even after system reboots, add this entry to the file /etc/fstab on the client machine (wiki server).

```
root@wiki:/
[root@wiki /]# cat /etc/fstab
# This file is edited by fstab-sync - see 'man fstab-sync' for details
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot             /boot                   ext3    defaults        1 2
/dev/md0                /data                   ext3    defaults        1 2
none                    /dev/pts                devpts  gid=5,mode=620  0 0
none                    /dev/shm                tmpfs   defaults        0 0
none                    /proc                   proc    defaults        0 0
none                    /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
/dev/hda                /media/cdrom            auto    pamconsole,exec,noauto,manag
/dev/fd0                /media/floppy           auto    pamconsole,exec,noauto,manag
172.20.35.74:/RPMS      /newmount               nfs     defaults        0 0
[root@wiki /]#
```

Check the last entry in the /etc/fstab file……
The entries added in /etc/fstab take effect after the following command.

#mount –a

15. Samba Server Configuration

Define samba?

Samba enables the users to share file systems and printers on network. Files residing in linux server can be accessed in windows machines through samba. Even files can be shared with other unix systems through samba.

(i) Packages

rpm -ivh samba-3.0.10-1.4E.6.i386.rpm
rpm -ivh samba-common-3.0.10-1.4E.6.i386.rpm
rpm -ivh samba-swat-3.0.10-1.4E.6.i386.rpm
rpm -ivh samba-client-3.0.10-1.4E.6.i386.rpm

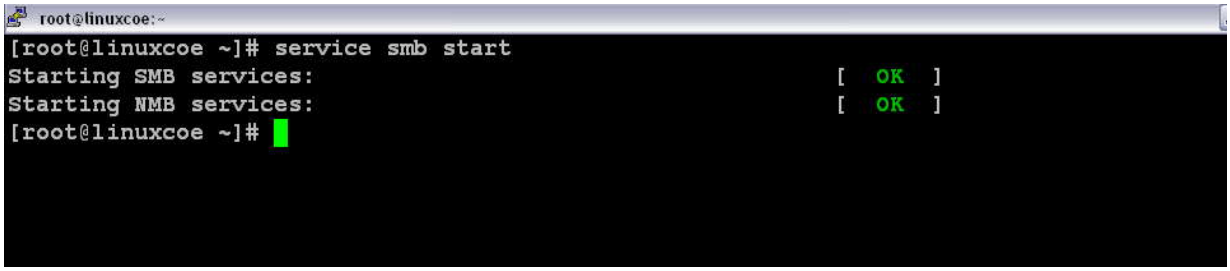
13. Daemons

smbd – Authentication ,Authorization, File & print sharing
nmbd - WINS Server, NetBIOS server & resource browsing

14. Starting the Samba Service

#service smb start

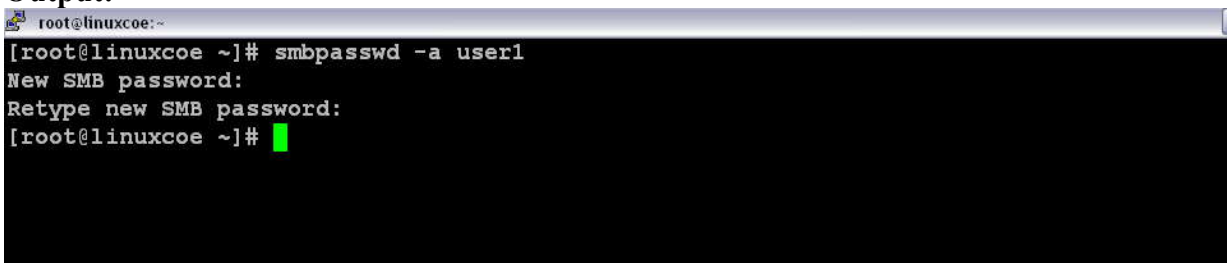
Output:



15. Setting samba passwd to the users

smbpasswd -a user1

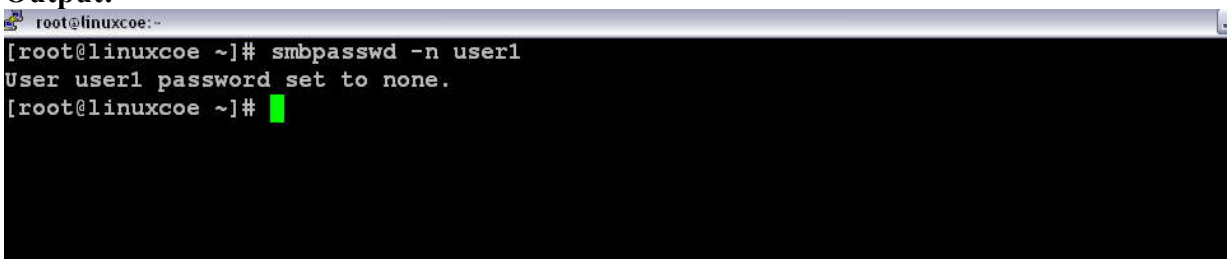
Output:



16. Removing samba password (if needed)

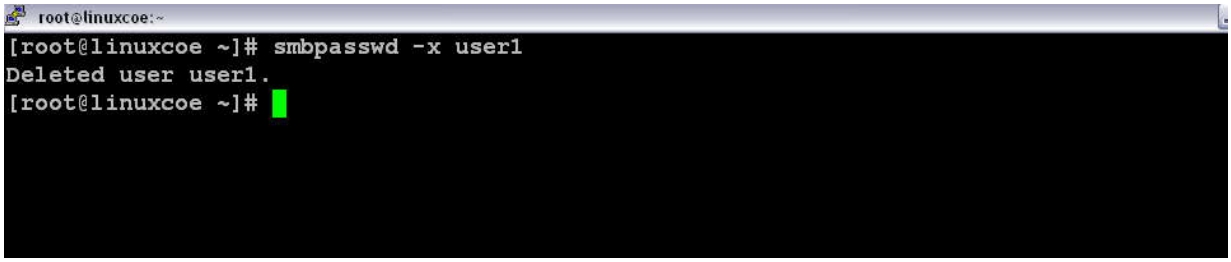
smbpasswd -n user1

Output:



17. Deleting Users in samba (if needed)

smbpasswd -x user1



18. Configuration File

/etc/samba/smb.conf

[ram]
Comment = user1s home directory
Path = /home/user1
Valid users = user1 user2
Public = no
Writable = yes
Printable = no
create mask = 0777.

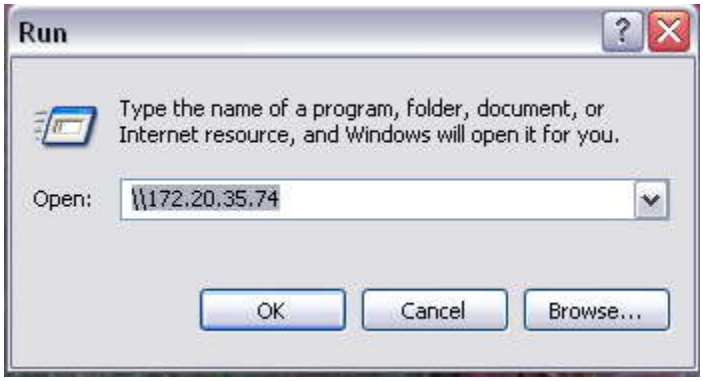
Type the above in the configuration file

```
[ram]
   comment = user1s home directory
   path = /home/user1/
   valid users = user1 user2
   readonly = No
   public = Yes
   writable = Yes
   printable = No
   create mask = 0755
```

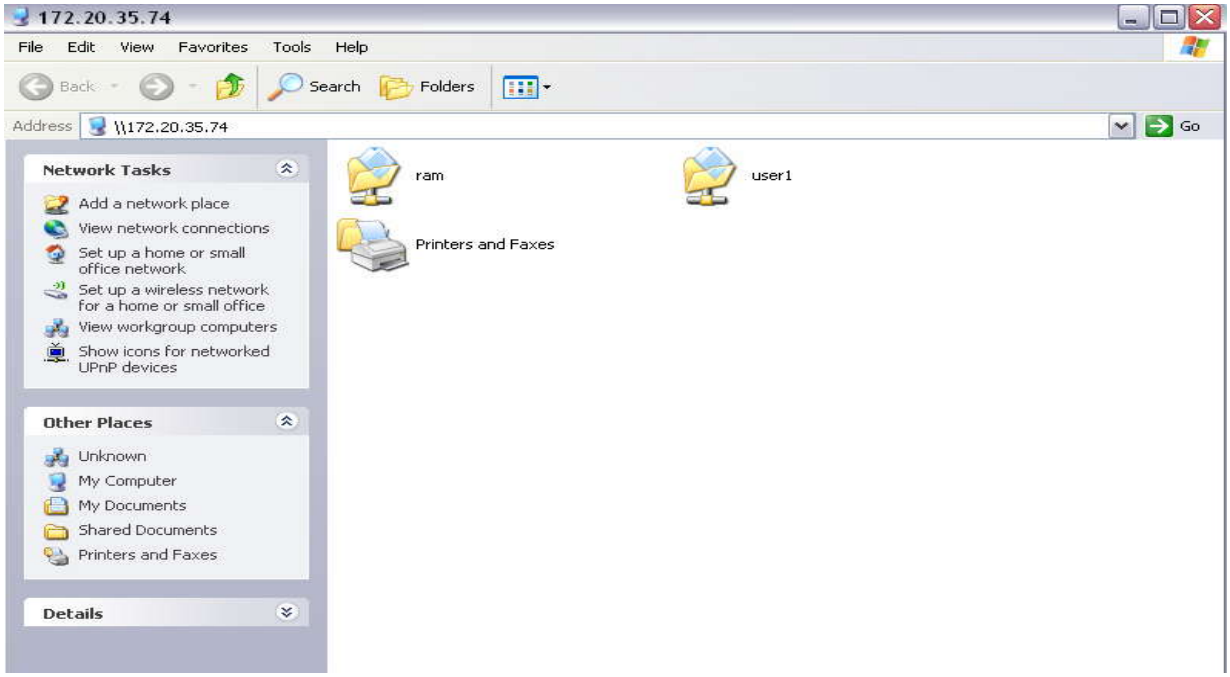
 # testparm

Testparm is a very simple test program to check an **smbd** configuration file for internal correctness. If this program reports no problems, you can use the configura tion file with confidence that **smbd** will successfully load the configuration file

19. Accessing Samba from Windows



Enter the username and password to access the samba share, and click o k.Then it will open the share.



20. Accessing Samba in other Unix Systems

```
# smbclient //172.20.35.74/ram  -U user1
```

Where

172.20.35.74 - is the ipaddress of the samba server
/ram - share name of the samba server
-U username - is the name of the samba user

16. NIS (NETWORK INFORMATION SERVICES)

1. Introduction:

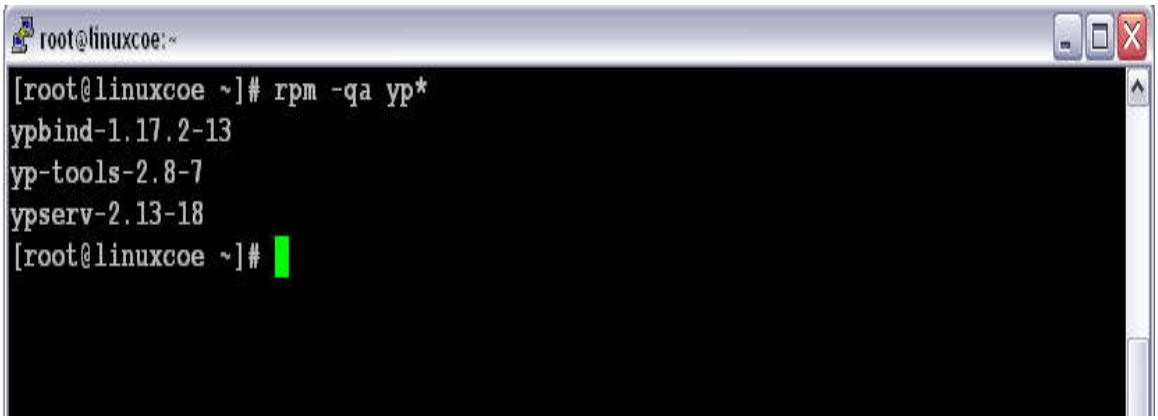
Network Information Services (NIS) enables you to create user accounts (as well as all system configuration files) that can be shared across all systems on your net work. The user account is created only on the NIS server.

2. Packages

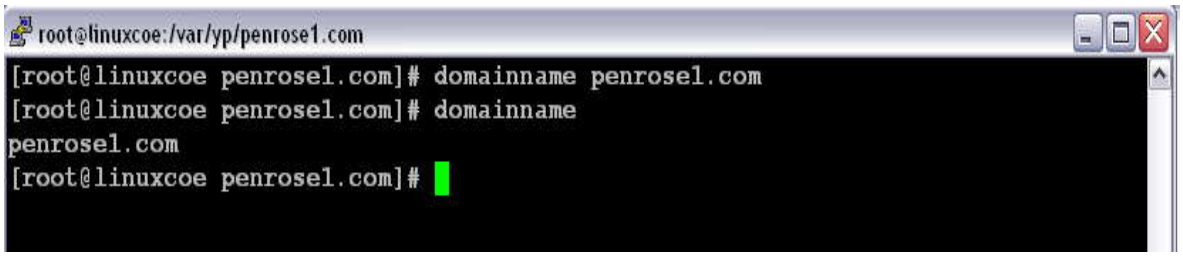
ypbind-1.19-7.el5.i386.rpm
ypserv-2.19-3.i386.rpm
yp-tools-2.9-0.1.i386.rpm

3. Configuring NIS Master Server

a. Check out whether NIS services are installed



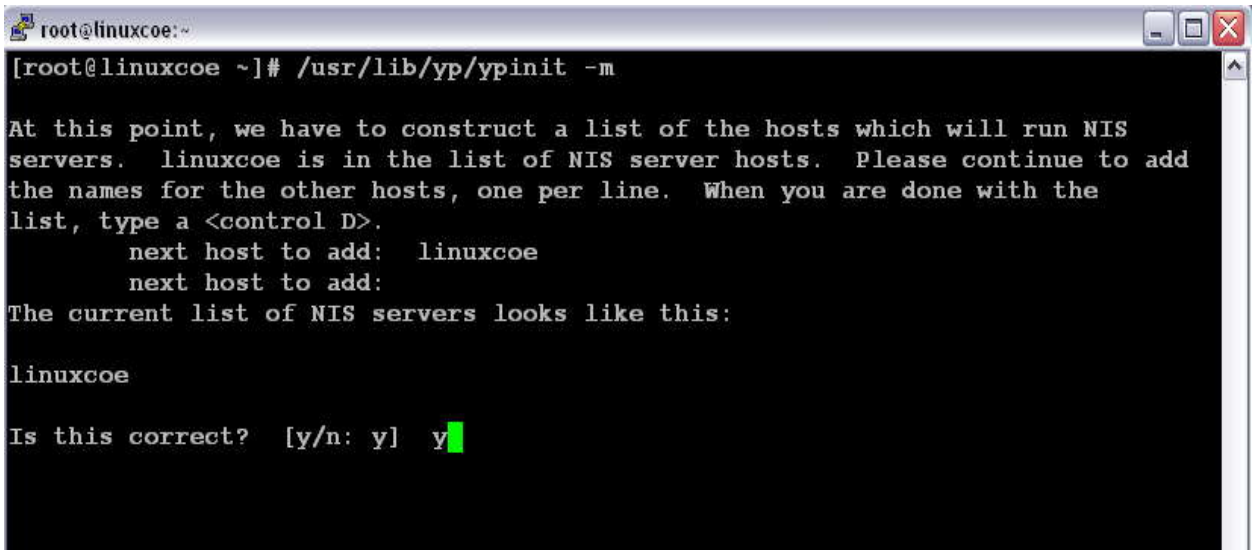
b. Set Domain Name to the system

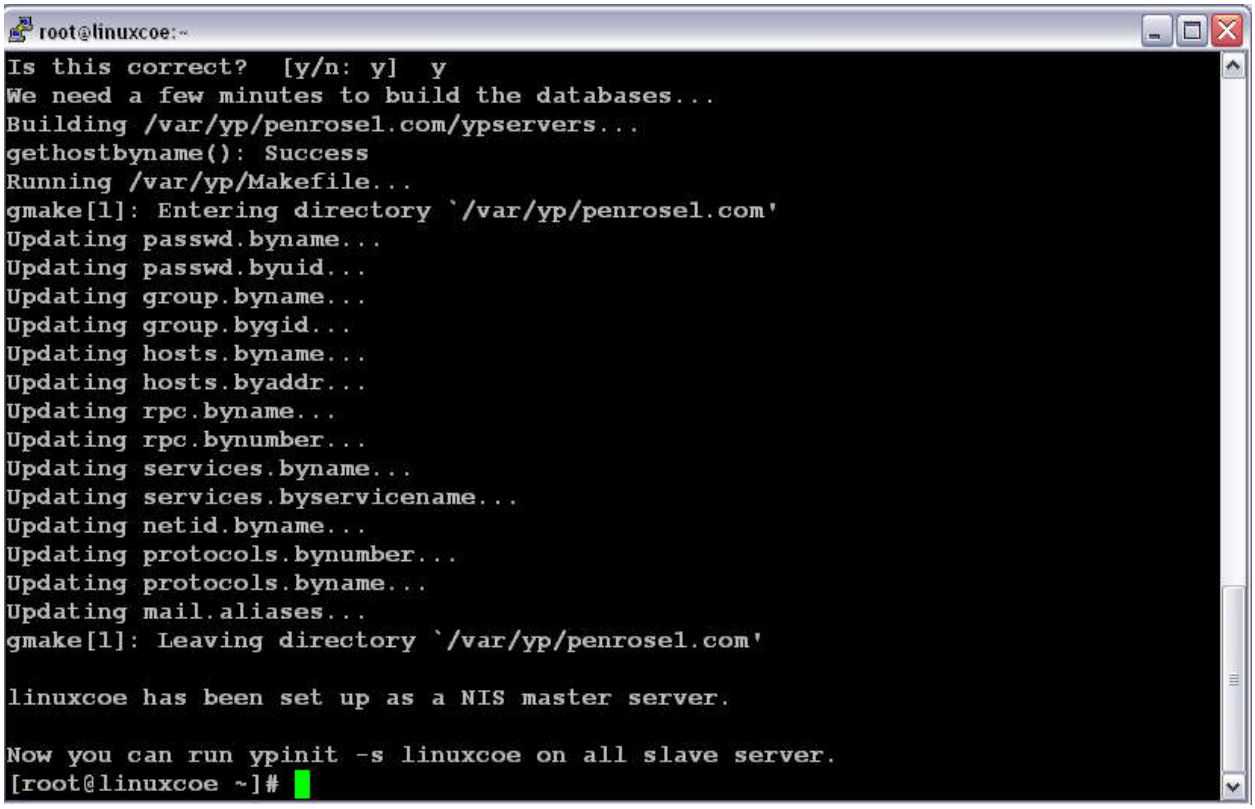


c. Create the NIS maps using the ypinit command

/usr/lib/yp/ypinit –m

<cntrl-d> & y to update

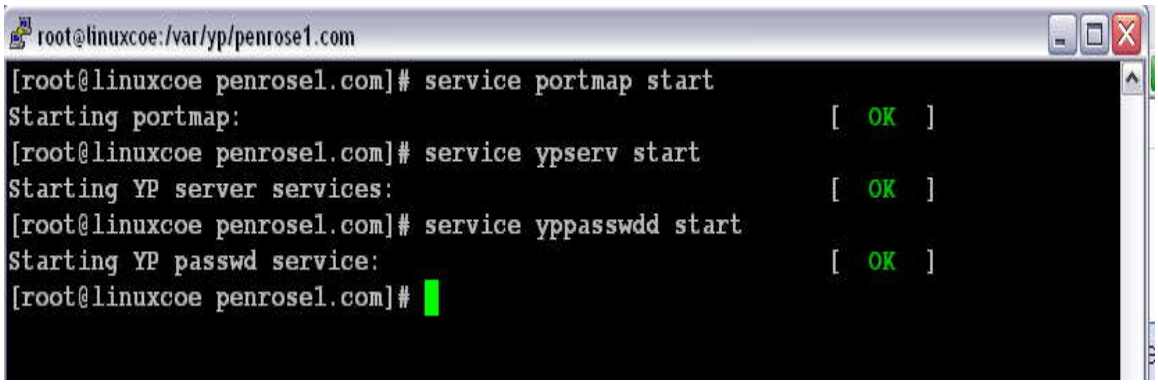




It creates the NIS master with the above configuration and updates to NIS database in the path /var/yp/penrose1.com.

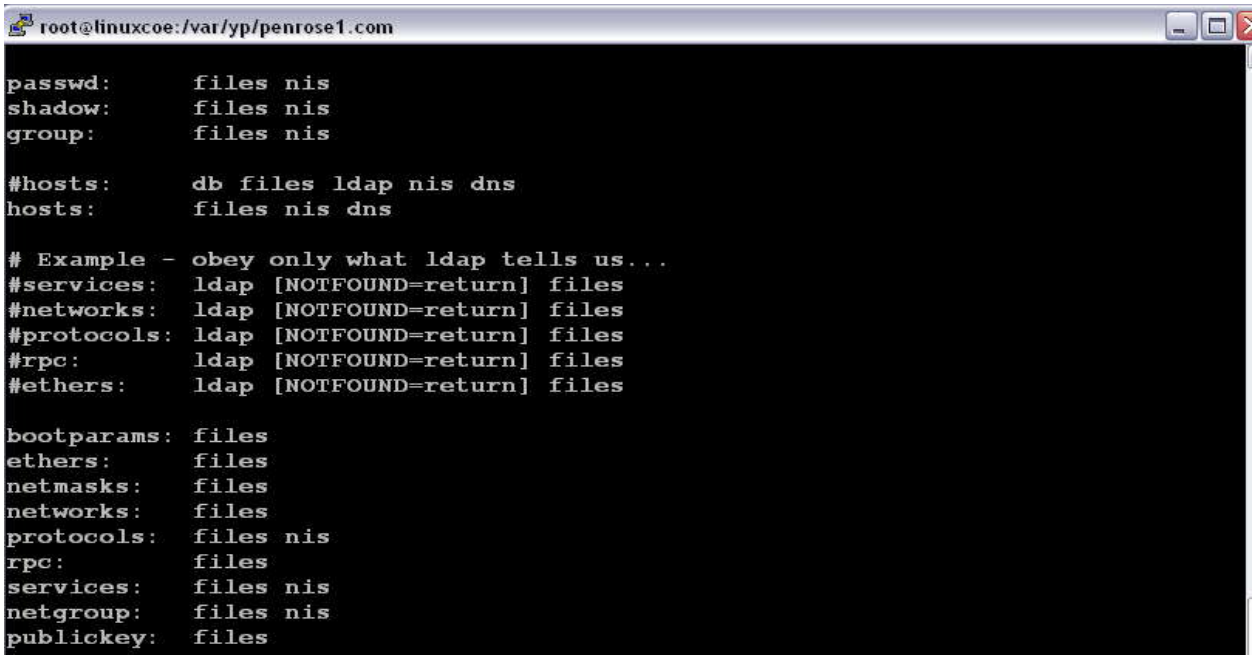
d. Start the following service in NIS server

- portmap service
- ypserv service
- yppasswdd service



Username and group name must be looked up in accordance with the /etc/nsswitch.conf configuration file.

The default to look them in the /etc/passwd file and the /etc/group file. In our Distributed Authentication System, the NIS users are looked up with a query to a NIS server (updates to NIS Database). Therefore in /etc/nsswitch.conf file set the lines as follows:



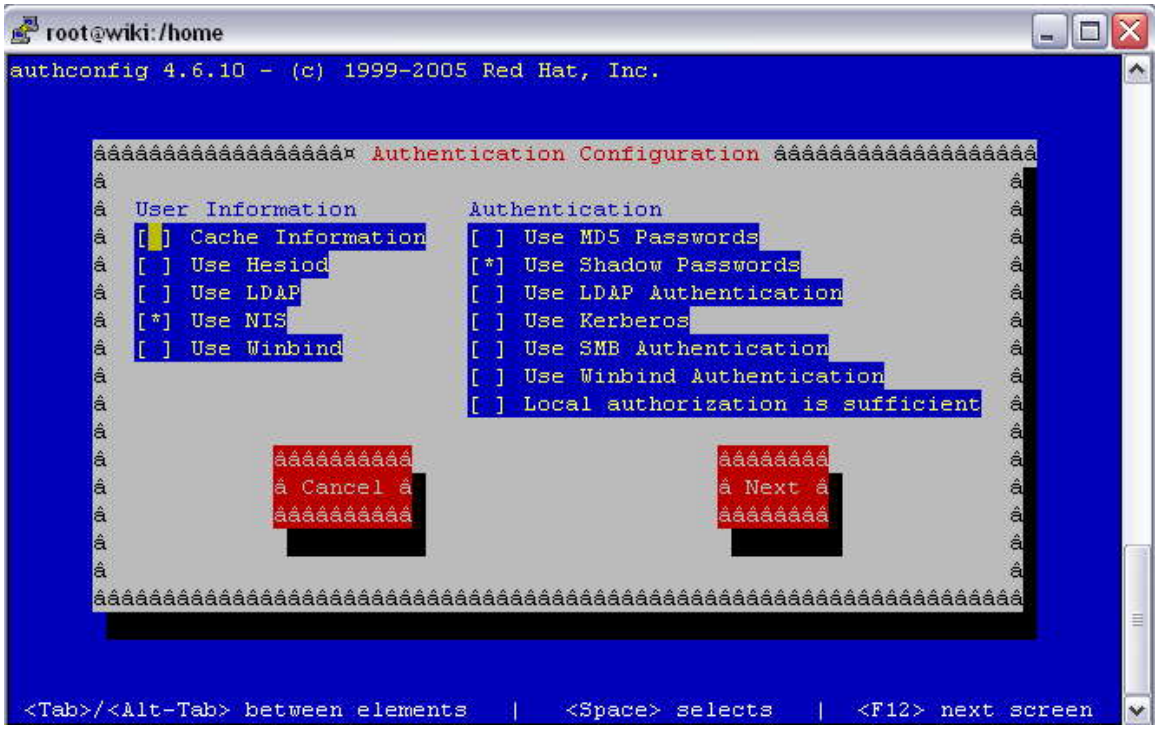
4. Configuring NIS Client

Do the above in /etc/nsswitch.conf file in client also.

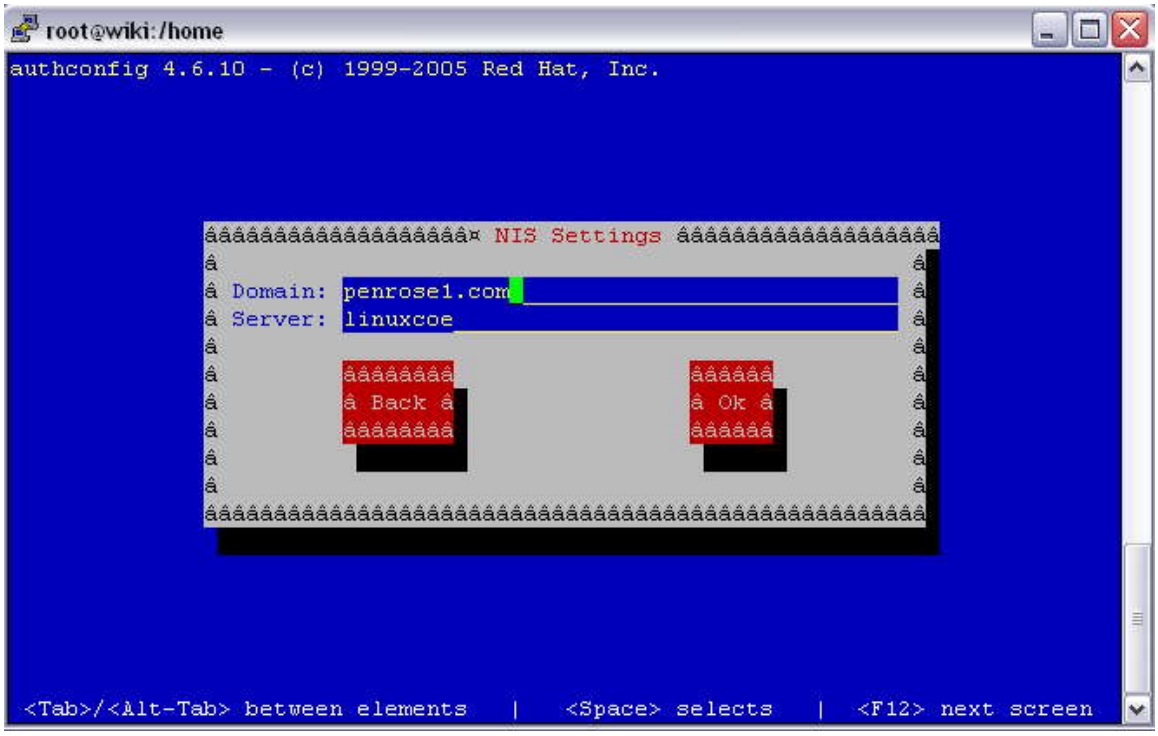
To set client to listen NIS Server, type

#authconfig (in RHEL5, use authconfig-tui)

It will open the following window; you have to select NIS as shown below

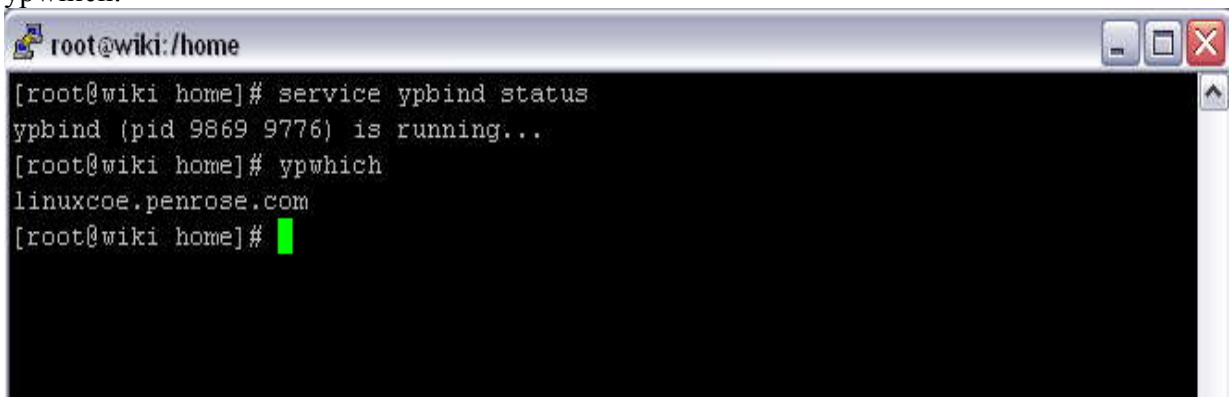


Then click next, give NIS Server Domain name and Server IP or Server hostname as follows:

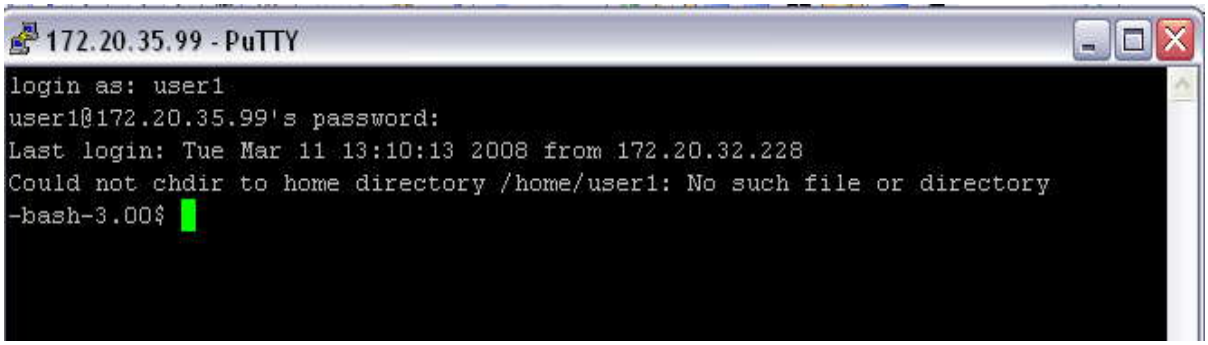


Run ypbind service in the client system, check out by

To check out the client communicating with NIS server, type the command ypwhich.



Thus now you can able to login as any user specified in NIS server in the client. For example, we are having an account user1 in NIS server, now we can login as user1 in the client system as follows:



The above screenshot shows that user1 can be logged in 172.20.35.99 system as NIS user, but it showing an error i.e. could not chdir to home directory /home/user1.

To solve that

In NIS Server

Edit the /etc/exports file to allow NFS mount the /home directory with read/write access

/home *(rw,sync)

Save the file and export it. Check out for nfs service running in the server.

IN NIS Client

IN NIS client system, start configuring autofs automounting. Edit your /etc/auto.master file to refer to file /etc/auto.home for mounting information whenever the /home directory is accessed.

#/etc/auto.master

/home /etc/auto.home --timeout 600

Add the line above mentioned in etc/auto.master file.

Edit file /etc/auto.home to do the NFS mount whenever the /home directory is accessed. If the line is too long to view on your screen, you can add a \ character at the end to continue on the next line.

```
#/etc/auto.home
*   -fstype=nfs,soft,intr,rsize=8192,wsize=8192,nosuid,tcp \
  192.168.1.100:/home:&
```

Start autofs and make sure it starts after the next reboot with the chkconfig command.

Then restart autofs service

```
#chkconfig autofs on
#service autofs restart
```

Now you will be able to login in remote system as NIS client to the server with the access of the directory of the user.

17. DNS - Domain Name System

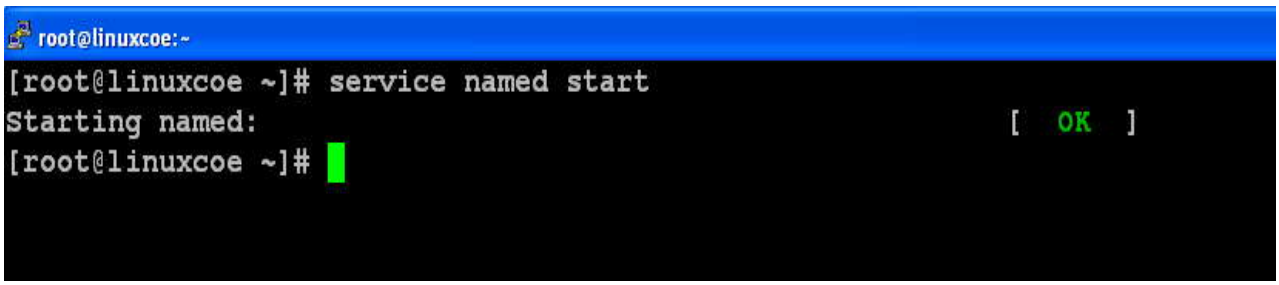
Domain Name System (or Service or Server), an Internet service that translates domain names into IP addresses.

Forward Lookup - Resolves hostnames into IP address
Reverse Lookup - Resolves IP addresses into hostnames

1. Packages

rpm -ivh bind-libs-9.2.4-24.EL4.rpm
rpm -ivh bind-devel-9.2.4-24.EL4.rpm
rpm -ivh bind-9.2.4-24.EL4.rpm
rpm -ivh bind-utils-9.2.4-24.EL4.rpm
rpm -ivh bind-chroot-9.2.4-24.EL4.rpm

2. Starting the DNS service



3. DNS Main configuration files

/etc/named.conf
/var/named/chroot/var/named/localhost.zone – forward file
/var/named/chroot/var/named/named.local – reverse file

4. Configuring DNS server

a. Editing named.conf file

```
root@linuxcoe:~
[root@linuxcoe ~]# vi /var/named/chroot/etc/named.conf
```

Changing the above file will be reflected in /etc/named.conf file.

b. For Forward Lookup

Add an entry in zone as penrose.com zone and mention the file name as Penrose.zone as shown below

```
root@linuxcoe:~
zone "localdomain" IN {
        type master;
        file "localdomain.zone";
        allow-update { none; };
};


zone "penrose.com" IN {
        type master;
        file "penrose.zone";
        allow-update { none; };
};

zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};
```

The zone name and file name may be anything, here we specified as penrose.com for zone and penrose.zone for the file.
Therefore there should be a file named penrose.zone in /var/named/chroot/var/named.
Copy the file localdomain.zone as penrose.zone in /var/named/chroot/var/named and give the entries as follows:

```
root@linuxcoe:/var/named/chroot/var/named
$TTL    86400
@               IN SOA  linuxcoe.penrose.com. root.penrose.com. (
                                        42              ; serial (d. adams)
                                        3H              ; refresh
                                        15M             ; retry
                                        1W              ; expiry
                                        1D )            ; minimum
                IN NS           linuxcoe.penrose.com
#localhost      IN A            127.0.0.1
wiki            IN A            172.20.35.99
~
~
```

c. Restarting the DNS service

```
root@linuxcoe:/var/named/chroot/var/named
[root@linuxcoe named]# service named restart
Stopping named:                                            [  OK  ]
Starting named:                                            [  OK  ]
[root@linuxcoe named]#
```

d. For Reverse Lookup

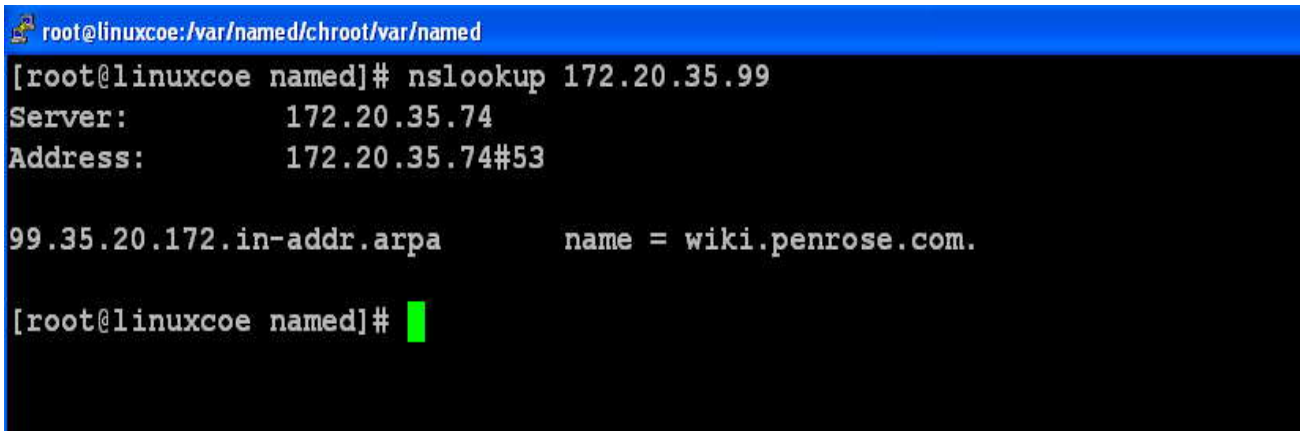
Add an entry in zone in-addr.arpa as “35.20.172.in-addr.arpa” for reverse lookup zone and mention the file name as penrose.local as shown below in /var/named/chroot/etc/named.conf

```
root@linuxcoe:/var/named/chroot/var/named
};

zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "named.local";
        allow-update { none; };
};


zone "35.20.172.in-addr.arpa" IN {
        type master;
        file "penrose.local";
        allow-update { none; };
};
```

Copy the file named.local as penrose.local in /var/named/chroot/var/named and give the entries as follows:

```
root@linuxcoe:/var/named/chroot/var/named
$TTL    86400
@       IN      SOA     linuxcoe.penrose.com. root.penrose.com.   (
                                        1997022700 ; Serial
                                        28800      ; Refresh
                                        14400      ; Retry
                                        3600000    ; Expire
                                        86400 )    ; Minimum
                IN      NS      linuxcoe.penrose.com.

1       IN      PTR     localhost.
99      IN      PTR     wiki.penrose.com.
~
~
```

For reverse lookup translation, now see the following output



5. Configuring DNS client

Edit the /etc/resolve.conf file and add the nameserver ip address as shown

Now restart the named service

6. To check whether DNS is running

```
root@linuxcoe:/var/named/chroot/var/named
[root@linuxcoe named]# nslookup wiki.penrose.com
Server:         172.20.35.74
Address:        172.20.35.74#53

Name:   wiki.penrose.com
Address: 172.20.35.99

[root@linuxcoe named]#
```

Therefore we have given wiki.penrose.com, it resolved it as 172.20.35.99 which is mentioned in the file penrose.zone.

18. LDAP

1. What is ldap

Lightweight Directory Access Protocol, a set of protocols for accessing information directories and runs over TCP/IP or other connection oriented transfer services.

2. Daemons

The OpenLDAP package includes two daemons: **slapd** and **slurpd**. The **slapd** daemon is the stand-alone LDAP daemon.

The **slurpd** daemon controls the replication of LDAP directories over a network.

3. Installation and Configuration

Required LDAP Server RPMS

openldap
openldap-clients
openldap-devel
nss_ldap
openldap-servers

Required LDAP Client RPMS

openldap
openldap-clients
openldap-devel
nss_ldap

4. Configuring The LDAP Server

The first stage of the project is to correctly configure the LDAP server. To do so, you must create an LDAP database and into which you import the /etc/passwd file

Create a database directory

For the example, create a dedicated example.com directory owned by the user ldap. (The ldap user is always created during the RPM installation process.)

[root@linuxcoe tmp]# mkdir /var/lib/ldap/example.com

```
[root@linuxcoe tmp]# chown ldap:ldap /var/lib/ldap/example.com
```

Create an LDAP "root" password

Only the LDAP root user can create, import data, and export data into an LDAP database. This user needs an encrypted password. You can create it with the slappasswd command and use the result in the LDAP configuration file.

```
[root@linuxcoe tmp]# slappasswd
New password:
Re-enter new password:
{SSHA}v4qLq/qy01w9my60LLX9BvfNUrRhOjQZ
[root@linuxcoe tmp]#
```

Edit the slapd.conf file

The LDAP server's daemon is named slapd and its configuration file is named /etc/openldap/slapd.conf. Update it with:

- A database of the default type bdb using the domain suffix example.com made up of domain components (DCs) example and com.
- The root user with a common name (CN), or nickname, of Manager who, as expected, is part of the example and com DCs.
- The encrypted version of the LDAP root password as well as the location of the LDAP database.

The configuration file syntax to do this is:

```
database      bdb
suffix        "dc=example,dc=com"
rootdn        "cn=Manager,dc=example,dc=com"
rootpw        {SSHA}v4qLq/qy01w9my60LLX9BvfNUrRhOjQZ
directory     /var/lib/ldap/example.com
```

Start the LDAP daemon

The service command uses the options start, stop, and restart to control the LDAP server's slapd daemon's operation. Use the start option to load the contents of the slapd.conf file.

Some LDAP versions require a DB_CONFIG file to be installed in the LDAP database directory. Before starting LDAP, you may need to copy a sample template file from /etc/openldap to /var/lib/ldap/example.com.

```
[root@linuxcoe tmp]# cp /etc/openldap/DB_CONFIG.example
/var/lib/ldap/example.com/DB_CONFIG
[root@linuxcoe tmp]# service ldap start
  Starting slapd: [  OK  ]
[root@linuxcoe tmp]#
```

Convert the /etc/passwd file to LDIF format

The data on the server's /etc/passwd file now needs to be converted to LDAP Data Interchange Files (LDIF) format before it can be imported into the LDAP database. Create the ldapuser test account

To create the ldapuser account you'll use for testing, type the commands.

```
[root@linuxcoe tmp]# useradd -g users ldapuser
[root@linuxcoe tmp]# passwd ldapuser
Changing password for user ldapuser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@linuxcoe tmp]#
```

Extract the desired records from /etc/passwd

You need to extract the ldapuser information from the /etc/passwd file using the grep command and save it by appending the information to the a file called /etc/openldap/passwd.ldapusers file with the > character.

```
[root@linuxcoe tmp]# grep ldapuser /etc/passwd > \
   /etc/openldap/passwd.ldapusers
[root@linuxcoe tmp]#
```

If this is your first time creating the LDAP database, you will also want to extract the information for the Linux root account from the /etc/passwd file to a brand new file called /etc/openldap/passwd.root.

```
[root@linuxcoe tmp]# grep root /etc/passwd > \
   /etc/openldap/passwd.root
[root@linuxcoe tmp]#
```

Find the conversion script

The /etc/passwd conversion program is called migrate_passw.pl; you can find it using the locate command. The locate utility updates its database every night and may not be able to find newly installed files.

```
[root@linuxcoe tmp]# locate -u
[root@linuxcoe tmp]# locate migrate
...
/usr/share/openldap/migration/migrate_passwd.pl
...
[root@linuxcoe tmp]#
```

Convert the ".ldapuser" file

You now need to convert the extracted /etc/passwd data into an LDIF that will then be imported into the database. Give the file used by regular users the name /etc/openldap/ldapuser.ldif and the one for the root user the name /etc/openldap/root.ldif.

```
[root@linuxcoe tmp]# /usr/share/openldap/migration/migrate_passwd.pl \
  /etc/openldap/passwd.ldapusers /etc/openldap/ldapusers.ldif
[root@linuxcoe tmp]#
[root@linuxcoe tmp]# /usr/share/openldap/migration/migrate_passwd.pl \
  /etc/openldap/passwd.root /etc/openldap/root.ldif
[root@linuxcoe tmp]#
```

Modify the LDIF files

With your two new LDIF files, the next step is to import this data into the LDAP database. To prepare for this, you must do some editing and create a new LDIF file that defines the organizational unit.

Edit the user LDIF file

migrate_passwd.pl script creates users that are all part of the organizational unit called People, but everyone belongs to the padl.com domain. You now have to edit both LDIF files and convert the string "padl" to "example" in each record. A text editor is fine for the job. For example, at the vi editor's : prompt, use the command:

```
%s/padl/example/g
```

to perform a global substitution of example for padl.

In the slapd.conf file, you gave the root user a common name (CN) of Manager. You now have to add this information to the root LDIF file by inserting this line under the UID line in the file.

cn: Manger

Create an LDIF file for the "example.com" domain

The LDIF files you created from /etc/passwd referred to users only. The attributes of the example.com domain haven't yet been defined, and you also haven't defined the organizational unit called People. This can be done using a third LDIF file called /etc/openldap/example.com.ldif,

```
dn: dc=example,dc=com
dc: example
description: Root LDAP entry for example.com
objectClass: dcObject
objectClass: organizationalUnit
ou: rootobject

dn: ou=People, dc=example,dc=com
ou: People
description: All people in organisation
objectClass: organizationalUnit
```

Import the LDIF files into the database

Use the LDAP add command to import all three LDIF files into the database starting with the example.com.ldif file, followed by root.ldif, and lastly by ldapusers.ldif.

Enter the LDAP root password you created when you are prompted.

```
[root@linuxcoe tmp]# ldapadd -x -D "cn=Manager,dc=example,dc=com" \
    -W -f /etc/openldap/example.com.ldif
[root@linuxcoe tmp]# ldapadd -x -D "cn=Manager,dc=example,dc=com" \
     -W -f /etc/openldap/root.ldif
[root@linuxcoe tmp]# ldapadd -x -D "cn=Manager,dc=example,dc=com" \
     -W -f /etc/openldap/ldapusers.ldif
[root@linuxcoe tmp]#
```

Test the LDAP database

You can view all the LDAP database entries all at once with the ldapsearch command; this is a good test to make sure you have all the correct functionality.

```
[root@linuxcoe tmp]# ldapsearch -x -b 'dc=example,dc=com' \
   '(objectclass=*)'
```

[root@linuxcoe tmp]#

5. Configuring the LDAP Client

Edit the ldap.conf configuration file

LDAP clients are configured using the /etc/openldap/ldap.conf file. You need to make sure that the file refers to the LDAP server's IP address for the domain example.com. The file should look like this:

HOST 192.168.1.100
BASE dc=example,dc=com

Edit the /etc/nsswitch file

The /etc/nsswitch.conf file defines the order in which the Linux operating system searches login databases for login information.

You want to configure it to first search its /etc/passwd file. If it doesn't find the user password information there, it goes to the LDAP server. The easiest way set this up is to use the /usr/bin/authconfig-tui command:

1. Run /usr/bin/authconfig-tui. The output of this command may be jumbled because your command line shell's language setting may not be compatible. You can usually avoid this problem by placing the string LANG=C in front of the command as shown here.

[root@smallfry tmp]# LANG=C authconfig-tui

1. Select LDAP.
2. Give the LDAP server's IP address, which is 192.168.1.100 in this case.
3. Give the base DN as dc=example,dc=com
4. Do not select TLS.
5. Use MD5 and shadow passwords.

The screen should look like this:

[*] Use Shadow Passwords
[*] Use MD5 Passwords
[*] Use LDAP [ ] Use TLS
Server: 192.168.1.100
Base DN: dc=example,dc=com

When finished, look at the /etc/nsswitch.conf file and make sure it has references to LDAP.

Note: In some Linux versions, the authconfig-tui command is replaced with the authconfig command.

Create Home Directories On The LDAP Client

You previously created a user named ldapuser in the group users on server linuxcoe. You now need to make sure that this user has a home directory on the LDAP client smallfry. The example in this section creates the directory and makes ldapuser the owner. As you can see, server smallfry correctly gets its user information about ldapuser from linuxcoe; the chown command doesn't complain about ldapuser not existing in smallfry's /etc/passwd file.

Check if ldapuser is Missing From the /etc/passwd file

You can look for ldapuser by searching the /etc/passwd file with the grep command. There should be no response.

```
[root@smallfry tmp]# grep ldapuser /etc/passwd
[root@smallfry tmp]#
```

Create The Home Directory For ldapuser On The LDAP Client

In this phase, you create the home directory, copy a BASH login profile file into it, and modify the ownership of the directory and all the files to user ldapuser.

Note: If the chown command fails, it is probably because of an incorrect LDAP configuration in which the LDAP client cannot read the user information from the LDAP server.

In some cases, you may want to use NFS mounts to provide home directories for your users, which will significantly reduce the need to do this step.

```
[root@smallfry tmp]# mkdir /home/ldapuser
[root@smallfry tmp]# chmod 700 /home/ldapuser/
[root@smallfry tmp]# ll /home
total 2
drwx------    2 ldapuser users       1024 Aug  4 08:05 ldapuser
[root@smallfry tmp]# cp /etc/skel/.* /home/ldapuser/
cp: omitting directory `/etc/skel/.'
cp: omitting directory `/etc/skel/..'
cp: omitting directory `/etc/skel/.kde'
[root@smallfry tmp]# chown -R ldapuser:users /home/ldapuser
[root@smallfry tmp]#
```

6. Configuring Encrypted LDAP Communication

There are two commonly mentioned methods of encrypting Linux LDAP communications between clients and servers. One method is through the use of the external stunnel utility that protects the data using SSL. The other method also uses SSL, but it is natively supported in LDAP by using its Transport Layer Security (TLS) option and is therefore easier to implement. This section describes both methods.

Using Transport Layer Security (TLS)Encryption

TLS is an updated version of the Secure Socket Layer (SSL) protocol used by many web browsers to do shopping cart checkouts. Like most certificate based encryption schemes it allows a client and server to talk in a trusted manner without the use of a password.

Configuring the TLS Server

1. Install the openssl-perl package which will provide the CA.pl script that helps to automate a lot of the certificate generation steps.

```
[root@linuxcoe tmp]# yum -y install openssl-perl
```

2. The location of the CA.pl script will vary with each Linux distribution. Use the updatedb command to update your file locations database to reflect the new files installed with the package and then use the locate command to file the script, which in this case is located at /etc/pki/tls/misc/CA.pl.

```
[root@linuxcoe tmp]# updatedb
[root@linuxcoe tmp]# locate CA.pl
/etc/pki/tls/misc/CA.pl
/usr/share/man/man1/CA.pl.1ssl.gz
[root@linuxcoe tmp]#
```

3. Take a look at the contents of the CA.pl script. There is a CATOP variable that determines where CA.pl will place some of the certificate files. With Fedora Core 5, this defaulted to the ../../CA directory. You could edit this to be a specific directory, but the script will get overwritten the original values the next time you update the packages on your system. In this example we'll do most of our work in a newly created /etc/openldap/TLS/data/files directory which will place the CA files in the /etc/openldap/TLS/CA directory.

```
[root@linuxcoe tmp]# mkdir -p /etc/openldap/TLS/data/files
[root@linuxcoe tmp]# cd /etc/openldap/TLS/data/files
[root@linuxcoe files]#
```

4. This next step is very important. You need to know the hostname of your system, and it should also be reflected in DNS or the host files of all your TLS LDAP clients. Things will not work other wise. Here it is linuxcoe.my-web-site.org.

```
[root@linuxcoe files]# hostname
linuxcoe.my-web-site.org
[root@linuxcoe files]#
```

5. Now you will run CA.pl for the first of many times. Create your CA certificate using CA.pl with the -newca option. You will be prompted for a PEM encryption pass phrase, use the same phrase for all the following TLS steps. The script will also prompt you for company in formation, if you don't use the defaults, make sure the information matches each time you run the script. Don't provide any of the "extra" attributes.

> **Note:** Make sure you place the correct hostname, not its IP address when you are prompted for it. The LDAP clients will only work if the hostname of the certificate matches the hostname, not IP address, of the server defined in their ldap.conf files.
> **Note:** Take note of your PEM encryption pass phrase. Your server default to a 365 day expiry at which point you will have to regenerate them from the CA certificate. If you don't remember the phrase, you'll have to start all over again.

```
[root@linuxcoe files]# /etc/pki/tls/misc/CA.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '../../CA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:
State or Province Name (full name) [Berkshire]:
Locality Name (eg, city) [Newbury]:
Organization Name (eg, company) [My Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:linuxcoe.my-web-site.org
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

```
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for ../../CA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            8f:e7:82:39:04:61:79:2b
        Validity
            Not Before: Jul  3 19: 02:39 2006 GMT
            Not After : Jul  2 19:02:39 2009 GMT
        Subject:
            countryName               = GB
            stateOrProvinceName       = Berkshire
            organizationName          = My Company Ltd
            commonName                = linuxcoe.my-web-site.org
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                FA:B4:14:1E:63:37:43:05:96:3E:E1:D7:24:9A:38:84:17:E4:A2:80
            X509v3 Authority Key Identifier:
                keyid:FA:B4:14:1E:63:37:43:05:96:3E:E1:D7:24:9A:38:84:17:E4:A2:80

Certificate is to be certified until Jul  2 19:02:39 2009  GMT (1095 days)

Write out database with 1 new entries
Data Base Updated
[root@linuxcoe files]#
```

Note: Your CA certificate will be named /etc/openldap/TLS/CA/cacert.pem. The private CA key used to generate all future server certificates will be named /etc/openldap/TLS/CA/private/cakey.pem. Guard this file carefully by making it only readable by the root user.

6. Run CA.pl again, but with the -newreq option to create a new server certificate signing request (CSR) and its associated server private key. Follow the same guidelines as before.

Your CSR will be named newreq.pem and its private key will be newkey.pem; both will be located in your current directory.

```
[root@linuxcoe files]# /etc/pki/tls/misc/CA.pl -newreq
```

Note: If the server certificate is going to be used on a different server running LDAP, then you'll have to use the hostname of that server when you are generating the request.

7. Run CA.pl one more time with the -sign option. The script will use the CSR information to create a server certificate signed by your CA certificate. You will be prompted for the CA private key's PEM pass phrase as a form of validation. Answer "y" whenever you are prompted to sign the certificate and commit it to the certificate database. Follow the same guidelines for the other prompts as before.

Your new signed certificate will be named newcert.pem.

```
[root@linuxcoe files]# /etc/pki/tls/misc/CA.pl -sign
```

8. The newkey.pem key file in its present form will always require the PEM password for it to be used with TLS. This can cause problems when it is used by programs like the

LDAP slapd daemon that don't take advantage of a keyboard. You will have to use the openssl command to remove this requirement.

```
[root@linuxcoe files]# openssl rsa -in newkey.pem -out newkey.pem
Enter pass phrase for newkey.pem:
writing RSA key
[root@linuxcoe files]#
```

9. Copy the files to their final resting place in the /etc/openssl/cacerts directory.

```
[root@linuxcoe files]# cp newkey.pem /etc/openldap/cacerts/slapd-key.pem
[root@linuxcoe files]# cp ../../CA/cacert.pem /etc/openldap/cacerts/
[root@linuxcoe files]# cp newcert.pem /etc/openldap/cacerts/slapd-cert.pem
```

Note: If the server certificate is going to be used on a different server running LDAP, then you'll have to copy these files to the same locations on that server.

10. Alter your LDAP server's slapd.conf file to make the TLS entries map to the newly created PEM files.

```
#
# File: slapd.conf
#

TLSCipherSuite          HIGH:MEDIUM:+SSLv2:+SSLv3:RSA
TLSCertificateFile      /etc/openldap/cacerts/slapd-cert.pem
TLSCertificateKeyFile   /etc/openldap/cacerts/slapd-key.pem
TLSCACertificateFile    /etc/openldap/cacerts/cacert.pem
TLSVerifyClient         allow
```

11. Change the permissions and ownership of the /etc/openldap/cacerts files so they are only readable by the slapd daemon that runs as the ldap user. Also secure the contents of the /etc/openldap/TLS directory from prying eyes.

```
[root@linuxcoe files]# chown ldap:ldap /etc/openldap/cacerts/*
[root@linuxcoe files]# chmod 600 /etc/openldap/cacerts/*
[root@linuxcoe files]# chown root:root /etc/openldap/TLS
[root@linuxcoe files]# chmod 750 /etc/openldap/TLS
```

12. Restart the LDAP daemon.

```
[root@zippy files]# service ldap restart
```

7. Configuring the TLS Client

Configuration of the client is much quicker as you will soon see. Here are the steps:

1. Run authconfig-tui and make sure your options match these screens.

```
------------------- Authentication Configuration -------------------
|                                                                   |
|  User Information       Authentication                            |
|  [ ] Cache Information   [*] Use MD5 Passwords                    |
|  [ ] Use Hesiod          [*] Use Shadow Passwords                 |
|  [*] Use LDAP            [*] Use LDAP Authentication              |
|  [ ] Use NIS             [ ] Use Kerberos                         |
|  [ ] Use Winbind         [ ] Use SMB Authentication               |
|                     [ ] Use Winbind Authentication                |
|                     [ ] Local authorizatio n is sufficient        |
|                                                                   |
|           ----------                  --------                    |
|           | Cancel |                  | Next |                   |
|           ----------                  --------                    |
|                                                                   |
|                                                                   |
-------------------------------------------------------------------
------------------- LDAP Settings --------------------
|                                      |
|         [*] Use TLS                  |
|  Server: linuxcoe.my-web-site.org_____________________ |
| Base DN: dc=example,dc=com_____________________________ |
|                                      |
|         --------              ------         |
|         | Back |              | Ok |         |
```

```
|       --------              ------        |
|                                     |
|                                     |
-------------------------------------------------
```

2. Review the contents of /etc/ldap.conf and make sure they have the following entries. The host must match the hostname of the certificate.

```
#
# File: /etc/ldap.conf
#

host         linuxcoe.my-web-site.org
base         dc=example,dc=com
tls_cacertdir  /etc/openldap/cacerts
ssl          start_tls
```

3. Review the contents of /etc/openldap/ldap.conf and make sure they have the following entries. The host must match the hostname of the certificate.

```
#
# File: /etc/openldap/ldap.conf
#
BASE        dc=example,dc=com
HOST        linuxcoe.my-web-site.org
TLS_CACERT    /etc/openldap/cacerts/cacert.pem
TLS_REQCERT   allow
```

4. Copy the server's /etc/openldap/cacerts/cacert.pem file to /etc/openldap/cacerts/cacert.pem on the LDAP client.

```
[root@smallfry tmp]# scp -P 222 \
root@linuxcoe.my-web-site.org:/etc/openldap/cacerts/cacert.pem \
/etc/openldap/cacerts/cacert.pem
```

6. Test your configuration by logging into your LDAP client server via SSL with the ldapuser user's credentials

8. TLS Maintenance

The server certificate (slapd-cert.pem) will expire annually and will have to be regenerated every year. The CA certificate (cacert.pem) is valid for three years before having to be regenerated and recopied to all your LDAP clients. You can change these defaults in the CA.pl file by editing the $DAYS variable for the server certificate and the $CADAYS variable for the CA certificate. Make sure that $CADAYS is greater than $DAYS. Here is an example of alternative values.

```
$DAYS="-days 365";     # 1  year
$CADAYS="-days 1825";  # 5 years
```

9. Configuring the stunnel LDAP client

First, you configure the LDAP client to use stunnel.

1. Edit the ldap.conf file. You have to trick the LDAP client into thinking that the LDAP server is actually running locally as a daemon, so you need to set the HOST entry to localhost. You then configure the stunnel utility to intercept this traffic and relay it to the real LDAP server.

```
HOST localhost
BASE dc=example,dc=com
```

2. Create an stunnel user with the useradd command.

```
[root@smallfry tmp]# useradd stunnel
```

3. Edit the stunnel.conf configuration file in the /etc/stunnel directory, configuring it as shown.

```
#
# File: /etc/stunnel (LDAP Client)
#

# Configure stunnel to run as user "stunnel" placing temporary
# files in the /usr/var/run/stunnel/ directory

chroot = /home/stunnel
pid = /stunnel.pid
setuid = stunnel
setgid = stunnel

# Configure logging
debug = 7
output = /var/log/messages

# Use it for client mode
```

client = yes

Service-level configuration
[ldap]
accept = 389
connect = 192.168.1.100:636

At the very end of the file, notice that traffic on the LDAP TCP port 389 is specifically redirected to the LDAP server on TCP port 636 over the secure tunnel.

4. Start stunnel with the stunnel command.

[root@smallfry tmp]# stunnel

5. Check the log files, especially the last 100 lines of the error log file /var/log/messages, to make sure there are no errors. If there are errors, double check your stunnel configuration file for mistakes.

[root@smallfry tmp]# tail -100 /var/log/messages

6. Make sure stunnel runs on the next reboot. The script /etc/rc.local is run at the end of every boot sequence. Use the locate command to find out where the stunnel program is and then place your stunnel command in /etc/rc.local as shown.

Run stunnel for LDAP (Fedora file location)
/usr/sbin/stunnel

10. Configuring the stunnel LDAP server

After you configure the client, you're ready to set up stunnel on the LDAP server.

1. Create an stunnel user using the useradd command.

[root@linuxcoe tmp]# useradd stunnel

2. Edit the stunnel.conf configuration file located in the /etc/stunnel directory. Configure it as shown.

#
File: /etc/stunnel (LDAP Server)
#

Configure stunnel to run as user "stunnel" placing temporary
files in the /usr/var/run/stunnel/ directory

```
chroot = /home/stunnel/
pid = /stunnel.pid
setuid = stunnel
setgid = stunnel

# Some debugging stuff
debug = 7
output = /var/log/messages

# Use it for client mode
client  = no
cert = /usr/share/ssl/certs/stunnel.pem
key =  /usr/share/ssl/certs/stunnel.pem

# Service-level configuration
[ldap]
accept  =  636
connect =  389
```

There are a few differences between the client and server stunnel.conf files. The very bottom of the file shows that all traffic received on the secure LDAP port of 636 is redirected to the application listening on LDAP port 389. The file is configured for server mode and a special SSH certificate has been defined for the encryption process. You'll create the certificates next.

3. Go to the /usr/share/ssl/certs directory and create the certificate using the make command. Use all the defaults when prompted, but make sure you use the server's IP address when prompted for your server's Common Name or hostname.

```
[root@linuxcoe tmp]# cd /usr/share/ssl/certs
[root@linuxcoe certs]# make stunnel.pem
...
Common Name (eg, your name or your server's hostname) []: 192.168.1.100
...
[root@linuxcoe certs]#
```

Note: The certificate created only has a 365 day lifetime. Remember to repeat this process next year.

4. Modify certificate file permissions. The certificate needs to be read by root and the stunnel user. Use the chmod and chgrp commands to do this.

```
[root@linuxcoe certs]# chmod 640 stunnel.pem
[root@linuxcoe certs]# chgrp stunnel stunnel.pem
[root@linuxcoe certs]# ll /usr/share/ssl/certs
 -rw-r-----  1 root stunnel   1991 Jul 31 21:50 stunnel.pem
[root@linuxcoe certs]#
```

5. Start stunnel with the stunnel command.

```
[root@linuxcoe tmp]# stunnel
```

6. Check the last 100 lines of the error log file /var/log/messages to make sure there are no errors. If you find errors, double check your stunnel configuration file for mistakes.

```
[root@linuxcoe tmp]# tail -100 /var/log/messages
```

The key things to look for are the loading of the certificate, the binding of LDAP to the 636 secure LDAP port, and the creation of the temporary stunnel.pid file.

```
2004.08.02 08:50:18 LOG7[12102:3210052320]: Certificate: /usr/share/ssl/certs/stunnel.pem
2004.08.02 08:50:18 LOG7[12102:3210052320]: Key file: /usr/share/ssl/certs/stunnel.pem
2004.08.02 08:50:18 LOG7[12102:3210052320]: ldap bound to 0.0.0.0:636
2004.08.02 08:50:18 LOG7[12103:3210052320]: Created pid file /stunnel.pid
```

7. Make sure stunnel runs on the next reboot. The script /etc/rc.local is run at the end of every boot sequence. Use the locate command to find out where the stunnel program is and then place your stunnel command in /etc/rc.local.

```
#
# File : /etc/rc.local
#
# Run stunnel for LDAP (Fedora file location)
/usr/sbin/stunnel
```

The final step of the preparation is to create home directories for each user to use just like in the unencrypted LDAP example before this. After this is complete, you'll need to do some basic testing which is covered in the troubleshooting section.

11. Common LDAP Administrative Tasks

Starting and Stopping LDAP

You can use the chkconfig command to get ldap configured to start at boot:

```
[root@linuxcoe tmp]# chkconfig ldap on
```

To start, stop, or restart ldap after booting, use

```
[root@linuxcoe tmp]# service ldap start
[root@linuxcoe tmp]# service ldap stop
[root@linuxcoe tmp]# service ldap restart
```

LDAP users changing their own passwords

LDAP users can modify their LDAP passwords using the regular passwd command.

```
[ldapuser@smallfry ldapuser]$ passwd
Changing password for user ldapuser.
Enter login(LDAP) password:
New password:
Retype new password:
LDAP password information changed for ldapuser
passwd: all authentication tokens updated successfully.
[ldapuser@smallfry ldapuser]$
```

Modifying LDAP users by user "root"

One easy way for the system administrator to manage LDAP users is to modify the regular Linux users' characteristics on the LDAP server in the regular way and then run a script to automatically modify the LDAP database.

The Modify LDAP User Script

You can use the very simple sample script /usr/local/bin/modifyldapuser to extract a particular user's information from /etc/passwd and import it into your LDAP database.

The script works by using the grep command to extract the /etc/passwd user record to a temporary file. It then runs the migrate_passwd script on this data and outputs the result to a temporary LDIF file. Next, the script replaces the default padl DC with the example DC and exports this to the final LDIF file. Finally, the ldapmodify command does the update, and then the temporary files are deleted.

```
#!/bin/bash

grep $1 /etc/passwd > /tmp/modifyldapuser.tmp

/usr/share/openldap/migration/migrate_passwd.pl \
  /tmp/modifyldapuser.tmp /tmp/modifyldapuser.ldif.tmp

cat /tmp/modifyldapuser.ldif.tmp | sed s/padl/example/ \
  > /tmp/modifyldapuser.ldif
```

```
ldapmodify -x -D "cn=Manager,dc=example,dc=com" -W -f \
   /tmp/modifyldapuser.ldif

rm -f /tmp/modifyldapuser.*

[root@linuxcoe tmp]# chmod 700 /usr/local/bin/modifyldapuser
[root@linuxcoe tmp]#
```

To use the script, modify the Linux user. In this case, modify the password for user ldapuser by running the modifyldapuser script using ldapuser as the argument. You will be prompted for the LDAP root password.

```
[root@linuxcoe tmp]# passwd ldapuser
Changing password for user ldapuser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@linuxcoe tmp]# modifyldapuser ldapuser
Enter LDAP Password:
modifying entry "uid=ldapuser,ou=People,dc=example,dc=com"
[root@linuxcoe tmp]#
```

12. Adding new LDAP users

Create an LDAP Add User Script

You can create a /usr/local/bin/addldapuser script based on the modifyldapuser script you created earlier. For example:

```
#!/bin/bash

grep $1 /etc/passwd > /tmp/changeldappasswd.tmp

/usr/share/openldap/migration/migrate_passwd.pl \
   /tmp/changeldappasswd.tmp /tmp/changeldappasswd.ldif.tmp

cat /tmp/changeldappasswd.ldif.tmp | sed s/padl/example/ \
   > /tmp/changeldappasswd.ldif

ldapadd -x -D "cn=Manager,dc=example,dc=com" -W -f \
   /tmp/changeldappasswd.ldif

rm -f /tmp/changeldappasswd.*
```

i. Add the User to the Database

Adding the user to database takes three steps:

1. Create the Linux user on the LDAP server.

2. Run the addldapuser script with the username as the only argument. This example imports a previously created Linux user named ldapuser. The script prompts you for your LDAP root password.

```
[root@linuxcoe tmp]# addldapuser ldapuser
Enter LDAP Password:
adding new entry "uid=ldapuser,ou=People,dc=example,dc=com"
[root@linuxcoe tmp]#
```

3. Create home directories for the user on all the LDAP client Linux boxes.

Remember that this script adds existing Linux users to the LDAP database. The creation of Linux users still requires the use of the adduser command.

ii. Deleting LDAP users

Sometimes you want to get rid of users instead of add them. You can create a /usr/local/bin/deleteldapuser script to delete LDAP users from your database. For example

```
#!/bin/bash

ldapdelete -x -W -D "cn=Manager,dc=example,dc=com" \
    "uid=$1,ou=People,dc=example,dc=com"
```

To delete the user from the database, run the deleteldapuser script with the username as the only argument. This example below deletes a previously created Linux user named ldapuser. The script prompts you for your LDAP root password.

```
[root@linuxcoe tmp]# deleteldapuser ldapuser
Enter LDAP Password:
```

19.DHCP

What is DHCP?

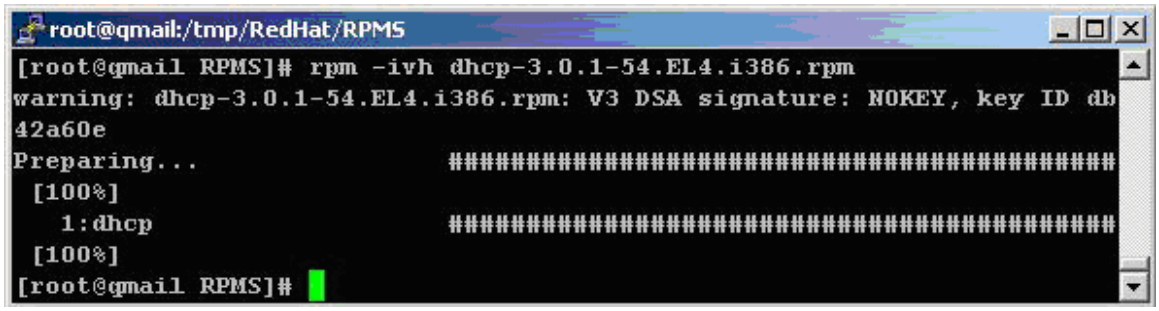
Dynamic Host Configuration Protocol, a protocol for assigning dynamic IP addresses to devices on a network. A new computer can be added to a network without the hassle of manually assigning it a unique IP address

1. Daemons

dhcpd -- Provides DHCP services to "lease" out IP addresses to remote machines.

2. Installing RPM Packages for DHCP

rpm –ivh dhcp-3.0.1rc14-1.i386.rpm



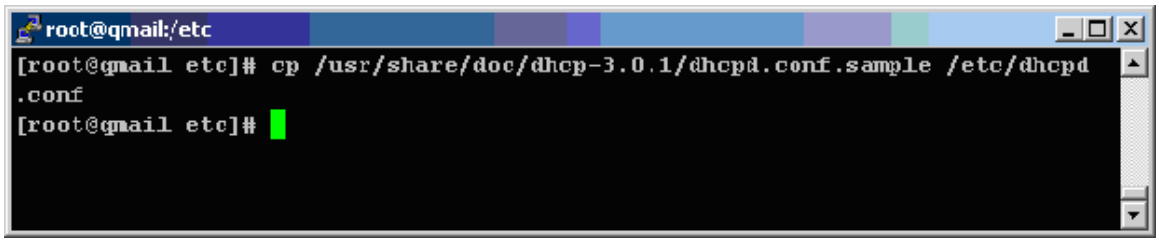
3. Configuring DHCP Server

- The configuration file is /etc/dhcpd.conf File
- The standard DHCP RPM package doesn't automatically install a /etc/dhcpd.conf file.

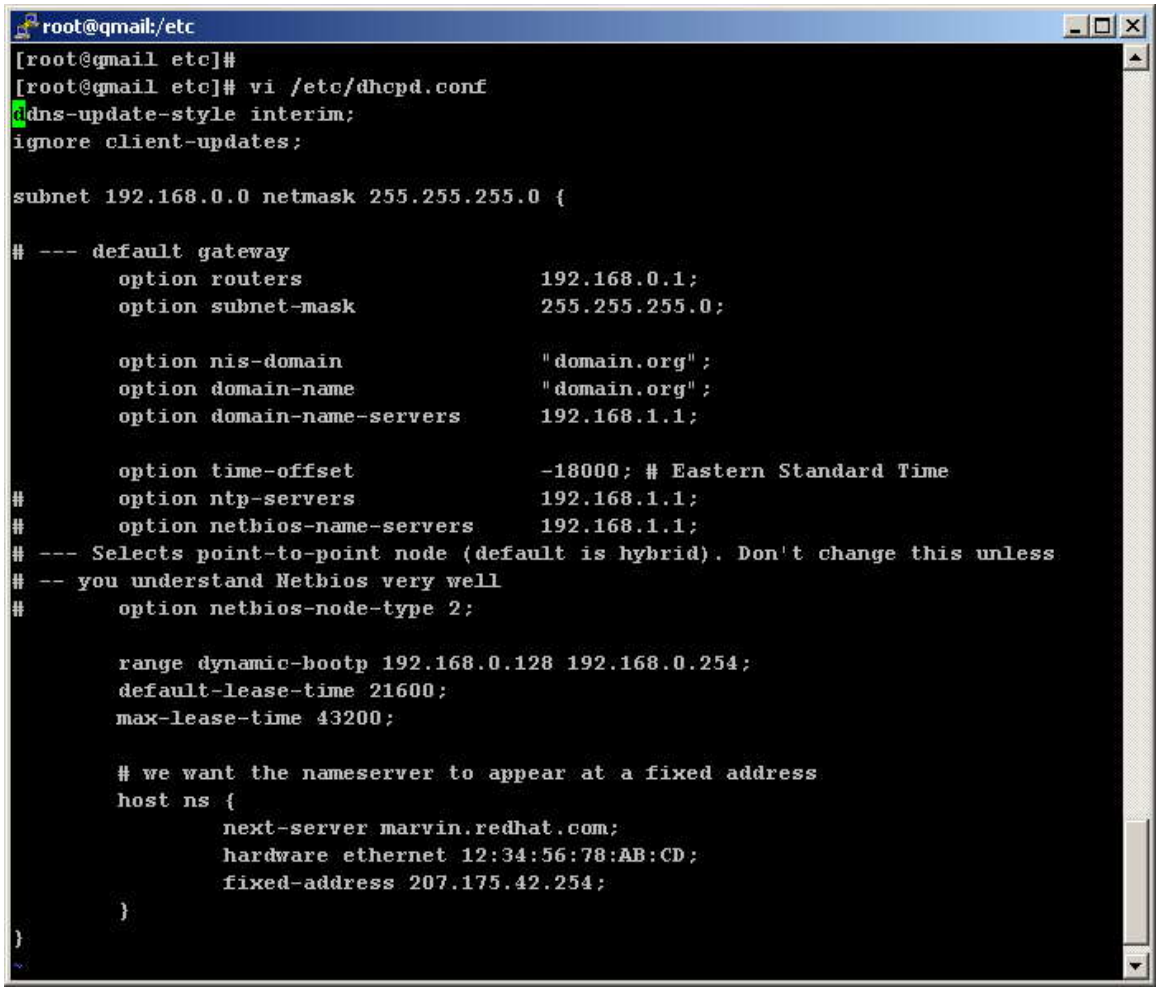
/usr/share/doc/dhcp-<version-number>/dhcpd.conf.sample

- Copy the sample dhcpd.conf file to the /etc directory and edit it.

[root@linuxcoe]# cp /usr/share/doc/dhcp-3.0pl1/dhcpd.conf.sample /etc/dhcpd.conf



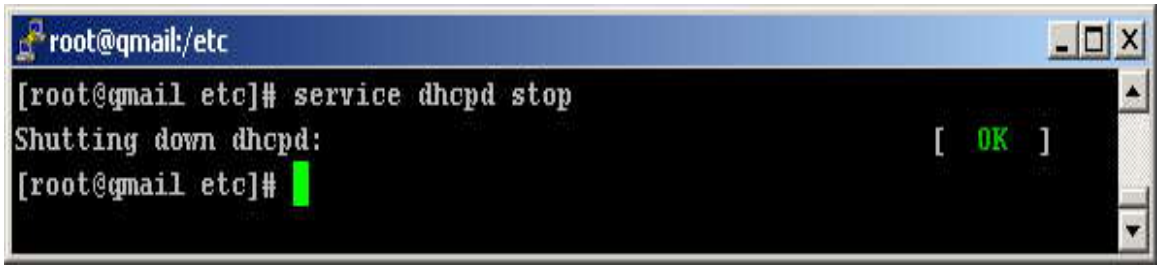
dhcpd.conf file



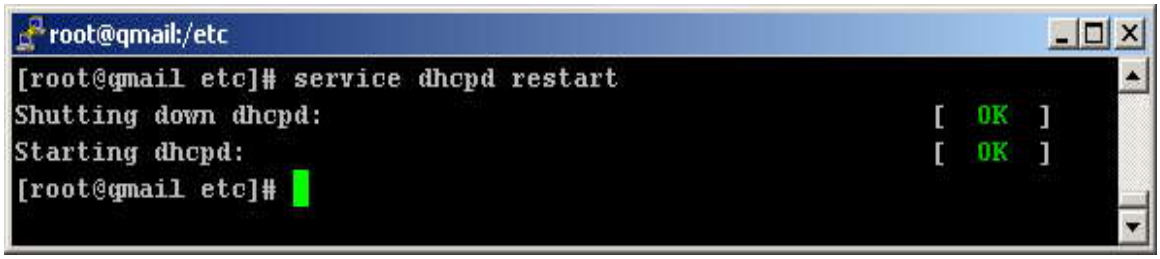
4. Starting / stopping / restarting the dhcpd services

#service dhcpd start

#service dhcpd stop



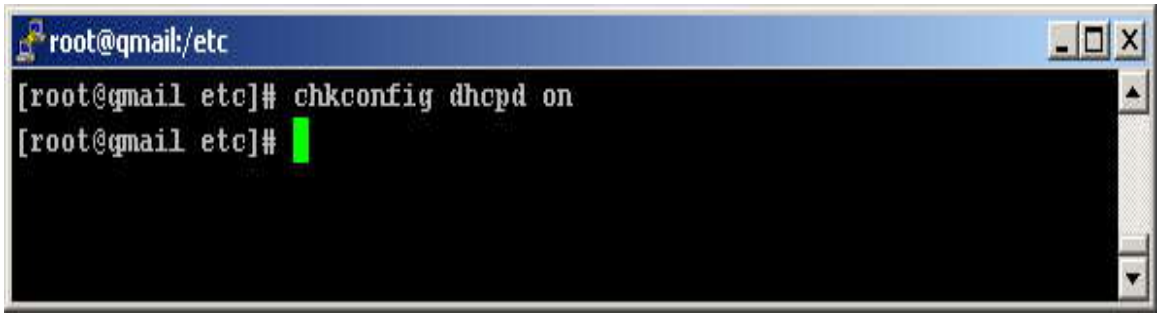
#service dhcpd restart



5. Getting DHCP Started

Use the chkconfig command to get DHCP configured to start at boot:

root@redhat# **chkconfig dhcpd on**



Test whether the dhcpd process is running with

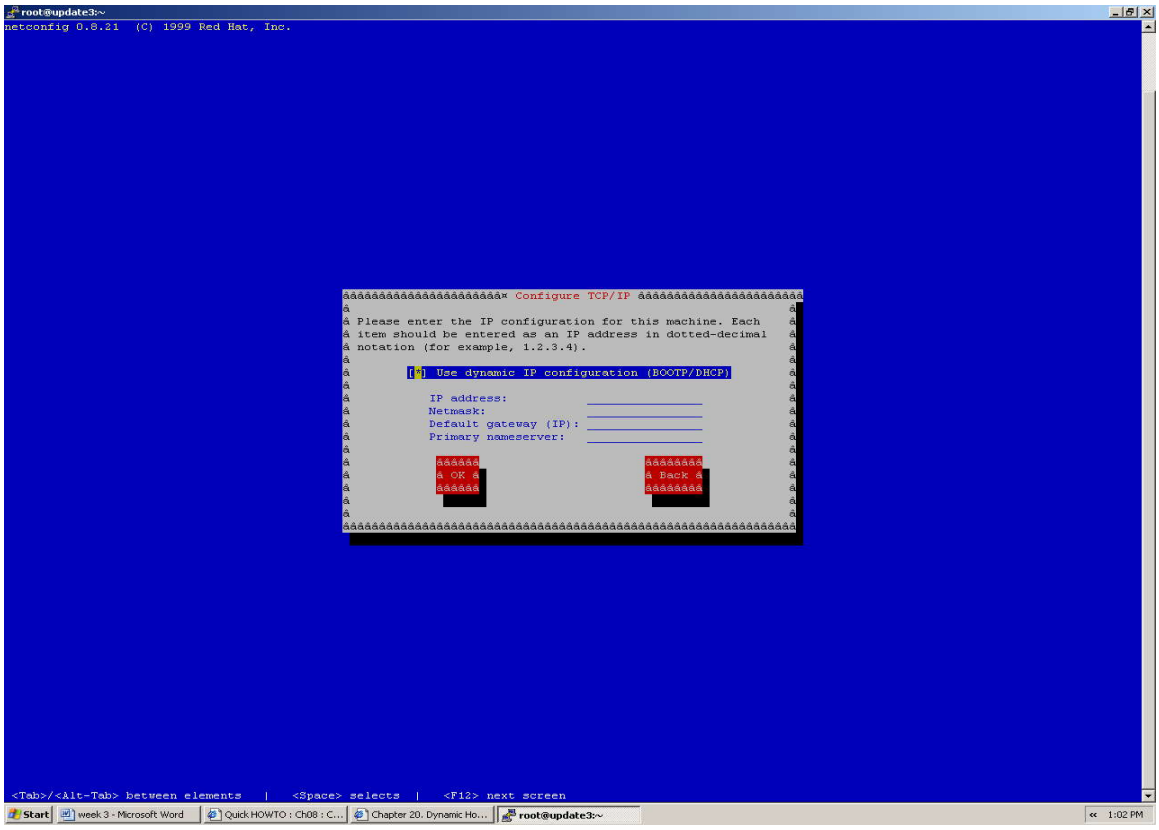
[root@redhat]# **pgrep dhcpd**

Finally set your PC to get its IP address via DHCP.

6. Configuring Linux and windows Clients to Use DHCP

A Linux NIC interface can be configured to obtain its IP address using DHCP. Windows defaults to using DHCP for all its NIC cards.

[root@update3 ~]# **netconfig**



Select dynamic IP configuration to assign IP from DHCP server.

FTP – File Transfer Protocol

FTP

A communication method for transferring data between computers on the Internet . FTP servers store files that can be accessed from other computers.

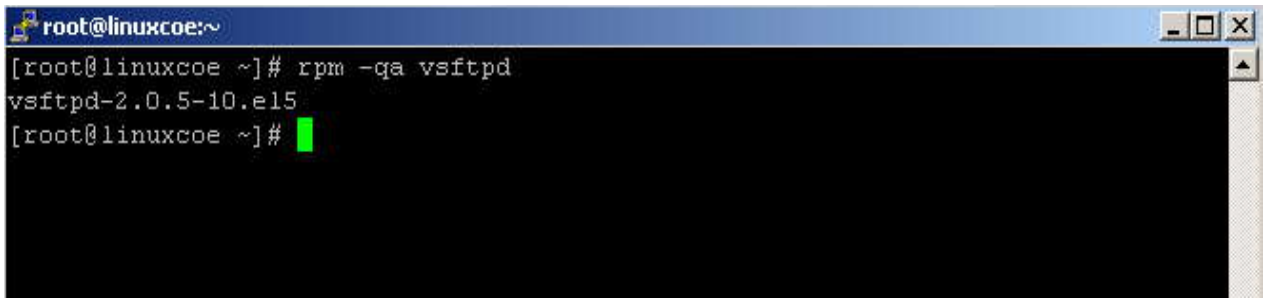
FTP provides security services so only authorized access is allowed.

1. **Ports**

FTP Control Channel, TCP Port 21: All commands user send and the ftp server's responses will go over the control connection, but any data sent back will go over the data connection.

FTP Data Channel, TCP Port 20: This port is used for all subsequent data transfers between the client and server.

2. **Check whether the RPM for VSFTPD service is installed**.



3. **Start the VSFTPD Service**

```
root@linuxcoe:~
[root@linuxcoe ~]# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@linuxcoe ~]#
```

4. **To start the service during system boot**

```
root@linuxcoe:~
[root@linuxcoe ~]# chkconfig vsftpd on
[root@linuxcoe ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@linuxcoe ~]#
```

5. **Configuration File**

The configuration file for FTP is **vsftpd.conf** which will be present in **/etc/vsftpd** directory

vsftpd directory also contains other important files such as **user_list, ftpusers**.

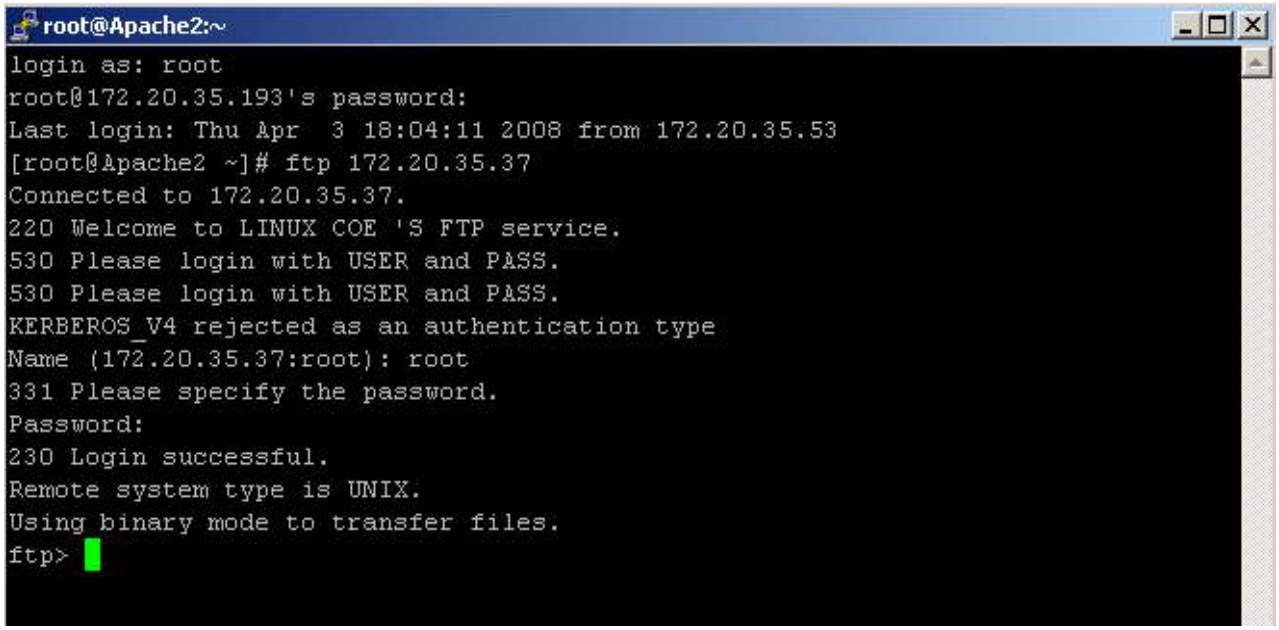
ftpusers This file contains the users who are not allowed to login via ftp. Therefore to login in ftp as root, root should be commented here.

6. **Logging in FTP Server**

Now open the ftp server in some other servers, so that we can transfer the data from the ftp server to the corresponding server system.

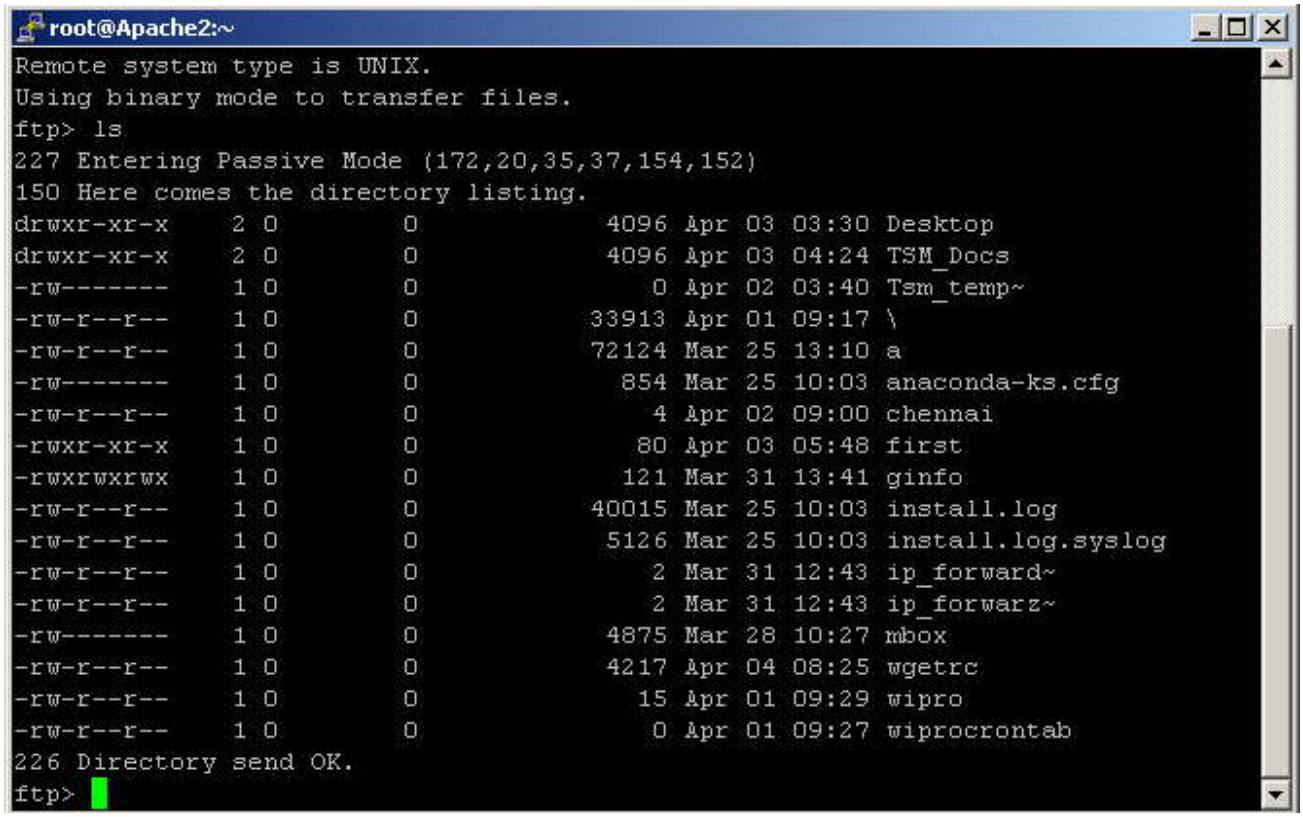
IP of linuxcoe system – 172.20.35.37

Type the following command in that server, it will authenticate according to the user (root or regular user or an anonymous user)

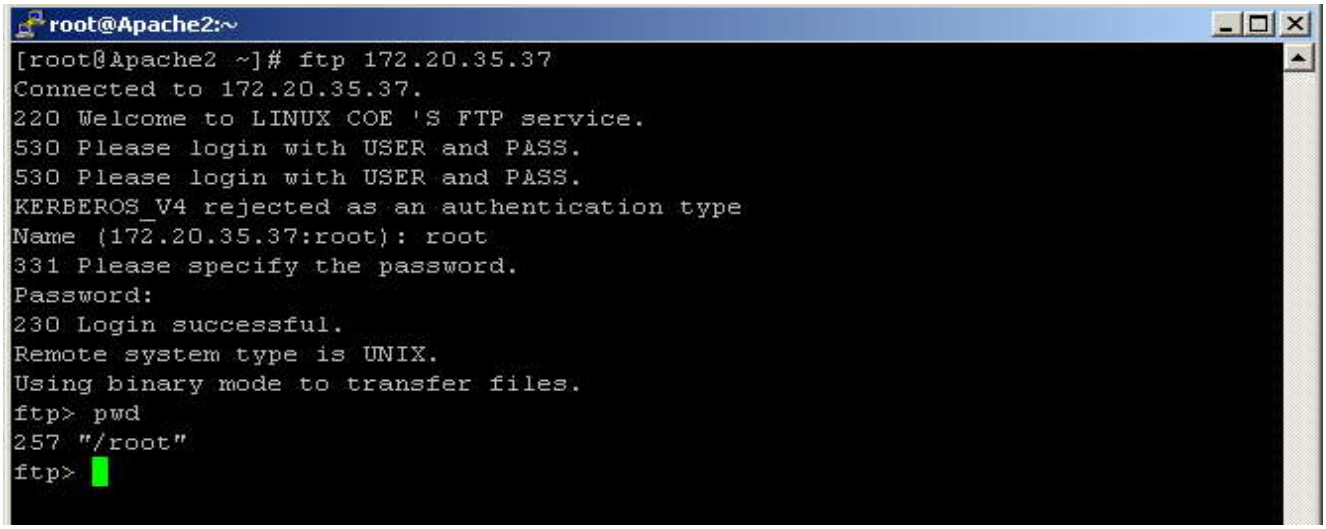


Therefore, now we have successfully logged in FTP server. So we can get files put files in FTP server.

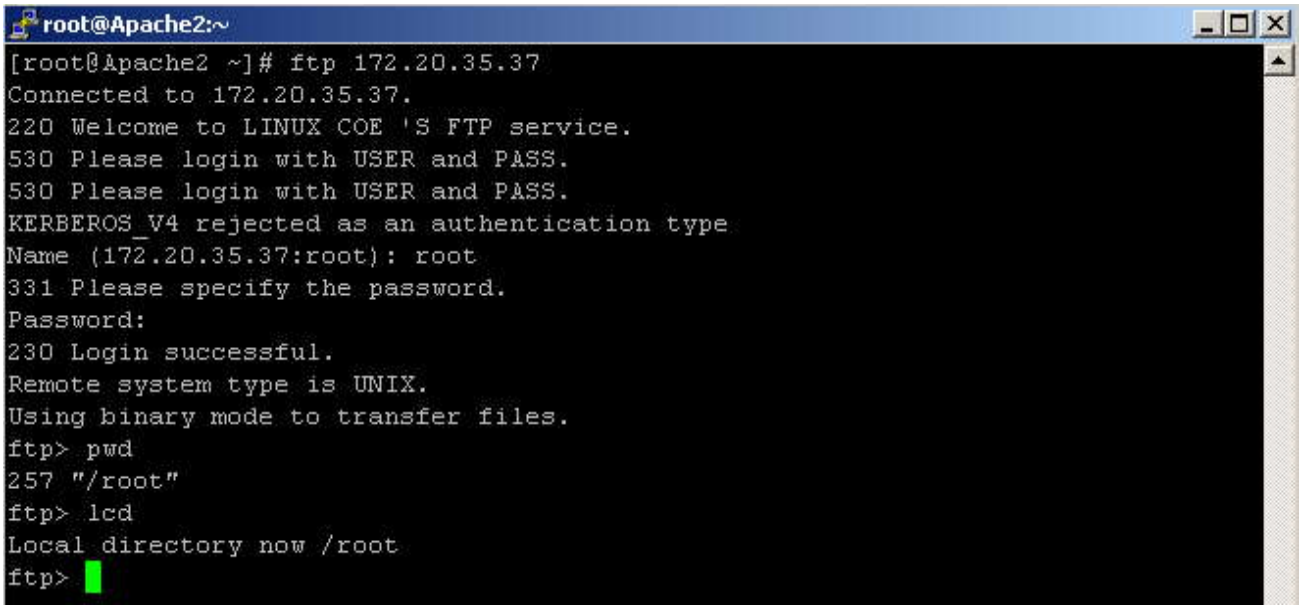
To list the files in FTP server:



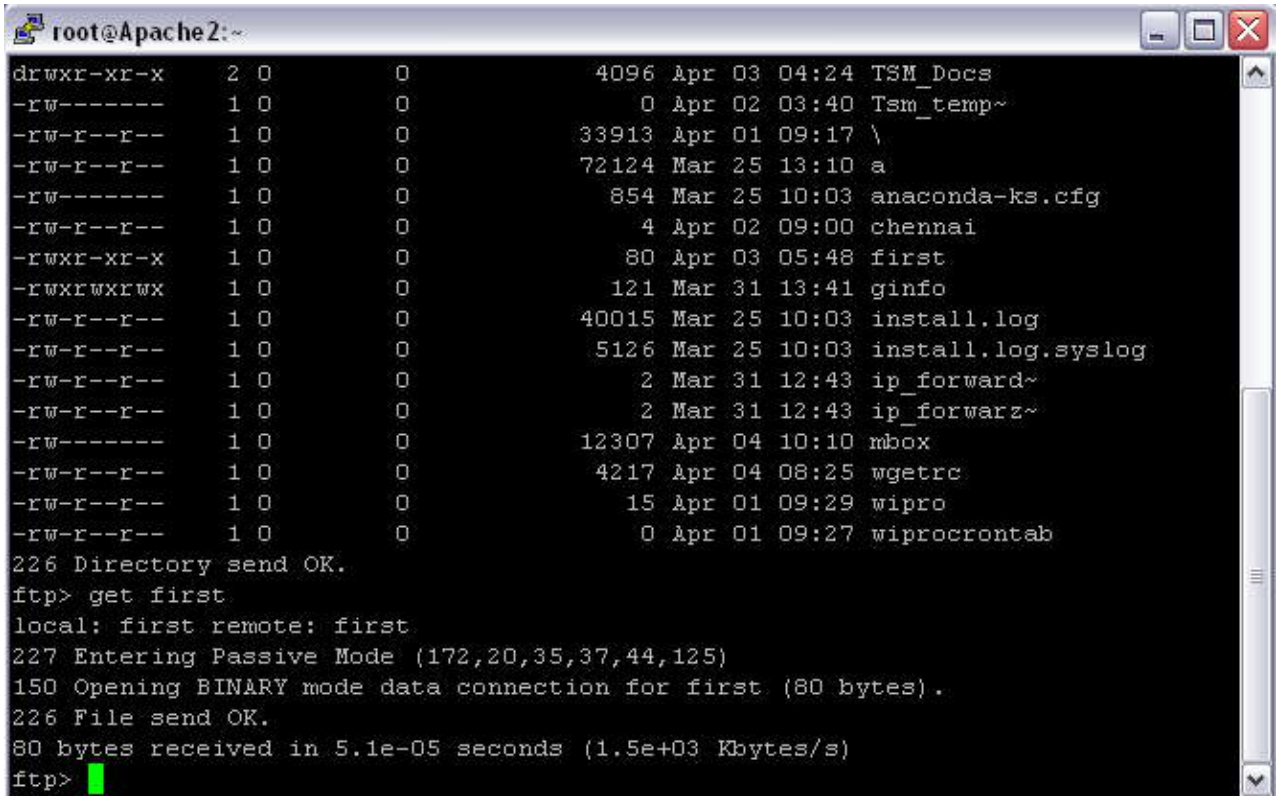
If we type **ls** there, we can see the contents of FTP server. But it is display the contents of the home directory of root because we logged in as root. We can see the current directory by the command **pwd.**



To see the local directory (the **pwd** of the apache server here), type the command **lcd**.

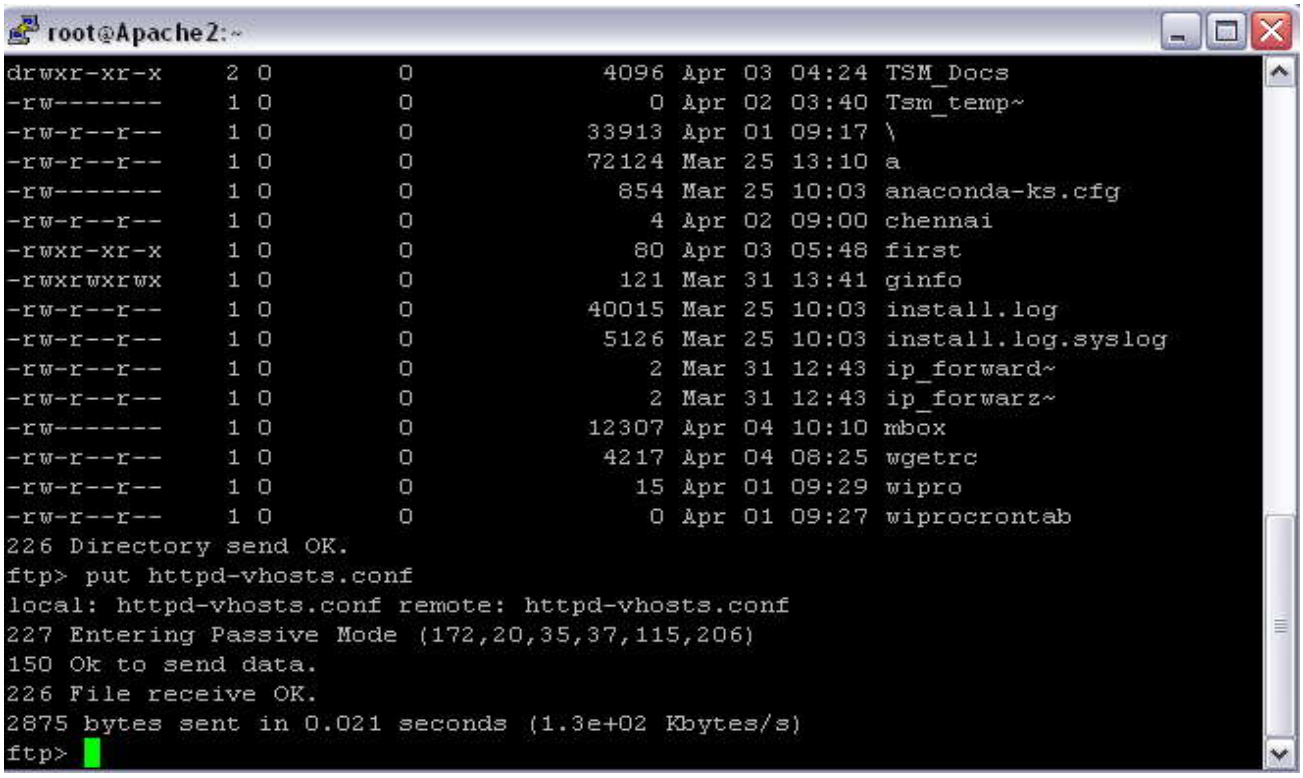


7. To get files from FTP Server:



Here, we got the file first from ftp server, the file first will be placed in the current directory. i.e. here in /root.

8. To put files in FTP Server:



Here we have put the file **httpd-vhosts.conf** from /root of the local directory to FTP server.

9. Anonymous Login

In **VSFTPD** configuration file, if we set **anonymous enable=YES**,. The default directory for anonymous user is **/var/ftp.**

20.Sendmail

What is Sendmail ?

Sendmail is the most popular Unix-based implementation of the simple Mail Transfer Protocol (SMTP) for transmitting e-mail.

1. **Daemons**

sendmail - allows to send emails using this machine as mail server

2. **Packages required**

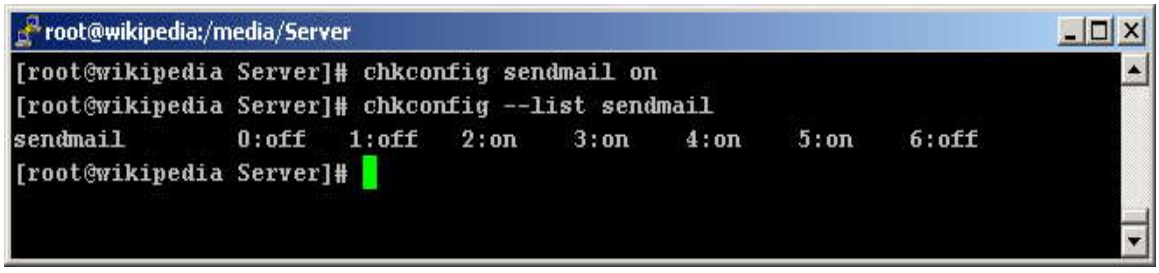
sendmail-8.12.10-1.1.1
sendmail-cf-8.13.1-2
sendmail-devel-8.13.1-2
dovecot-0.99.11-2.EL4.1
m4-1.4.1-16

3. **Installation and Configuration**

- # rpm -ivh sendmail-8.13.8-2.el5.i386.rpm
- # rpm -ivh sendmail-cf-8.13.8-2.el5.i386.rpm
- # rpm -ivh sendmail-devel-8.13.8-2.el5.i386.rpm
- # rpm -ivh sendmail-doc-8.13.8-2.el5.i386.rpm
- # rpm -ivh dovecot-1.0-1.2.rc15.el5.i386.rpm
- # rpm -ivh m4-1.4.5-3.el5.1.i386.rpm

4. **Starting Sendmail**

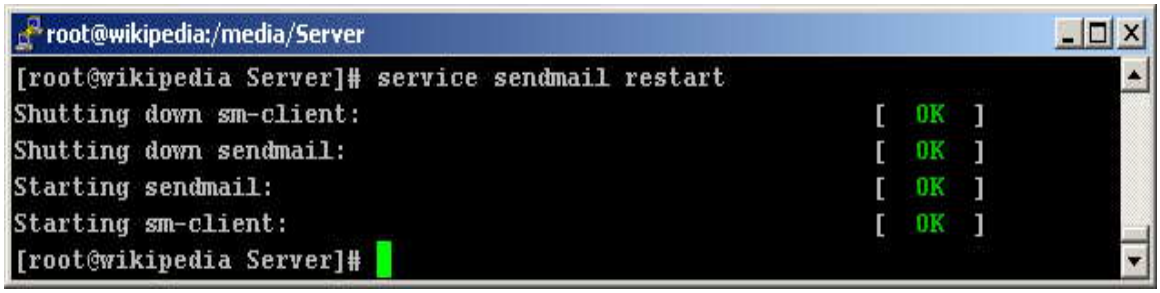
chkconfig sendmail on



5. **To start, stop, and restart sendmail after booting, use**

- # service sendmail start (to start sendmail service)

- # service sendmail stop (to stop sendmail service)
- # service sendmail restart



Script encapsulates all the required post configuration steps.

```
#!/bin/bash
cd /etc/mail
make
newaliases
/etc/init.d/sendmail restart
```

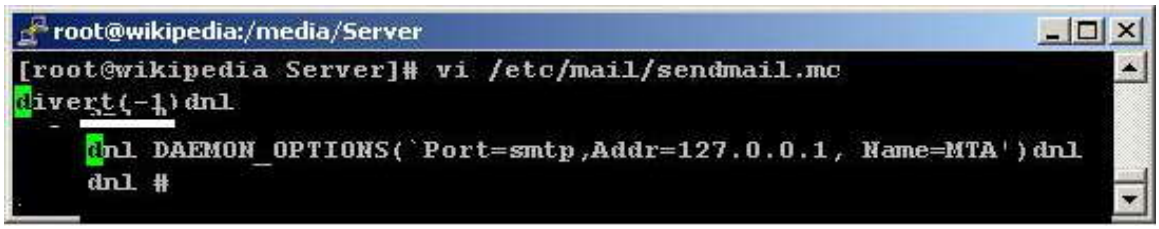
- It first runs the make command, which creates a new **sendmail.cf** file from the **sendmail.mc** file
- And compiles supporting configuration files in the /etc/mail directory according to the instructions in the file **/etc/mail/Makefile**.
- It then generates new e-mail aliases with the newaliases command and then restarts sendmail.

The **/etc/mail/sendmail.mc** File :

sendmail's configuration parameters in **the /etc/mail/sendmail.mc** file, which is then used by the m4 macros to create the **/etc/mail/sendmail.cf** file.

The sendmail.mc file doesn't use this character for commenting, but instead uses the string "dnl". Here are some valid examples of comments used with the sendmail.mc configuration file:

a. These statements are disabled by dnl commenting.
b. dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')
c. dnl # DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')
d. This statement is incorrectly disabled:
e. # DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')
f. This statement is active:
g. DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA'



6. Configuring DNS for sendmail

Configure Your Mail Server's Name In DNS

If mail server's name is linuxcoe and it you intend for it to mostly handle mail for the domain my- site.com, then linuxcoe.my-site.com must correctly resolve to the IP address of one of the mail server's interfaces. You can test this using the host command:

[root@linuxcoe]# host linuxcoe.my-site.com
linuxcoe.my-site.com has address 192.168.1.100
[root@linuxcoe]#

7. Configure The /etc/resolv.conf File

Sendmail program expects DNS to be configured correctly on the DNS server.

The first one is the **/etc/resolv.conf** file in which there must be a domain directive that matches one of the domains the mail server is expected to handle mail for.

For example, if the mail server is handling mail for my-site.com and the IP address of the DNS server is 192.168.1.100, there must be directives that look like this:

domain my-site.com
nameserver 192.168.1.100

The /etc/hosts file

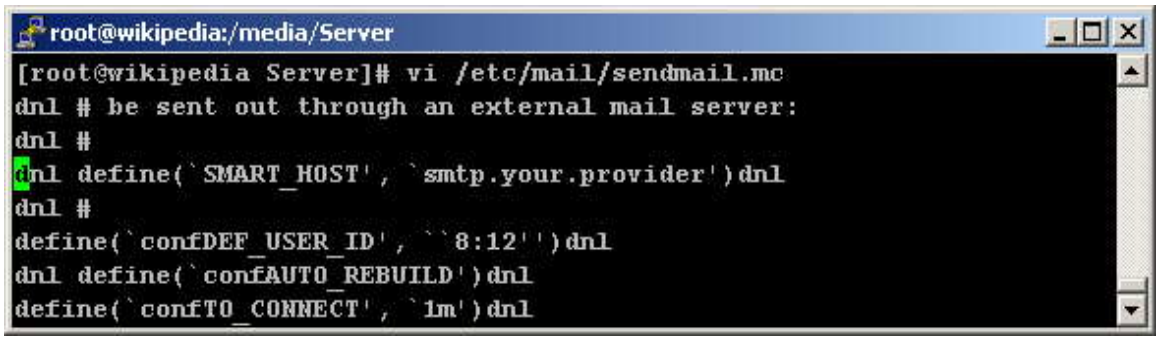
The **/etc/hosts** file also is used by DNS clients and also needs to be correctly configured.

127.0.0.1 linuxcoe.my-site.com localhost.localdomain localhost linuxcoe

8. Configure Linux Sendmail Clients

All Linux mail clients in your home or company need to know which server is the mail server. This is configured in the sendmail.mc file by setting the SMART_HOST statement to include the mail server. In the example below, the mail server has been set to mail.my-site.com, the mail server for the my-site.com domain.

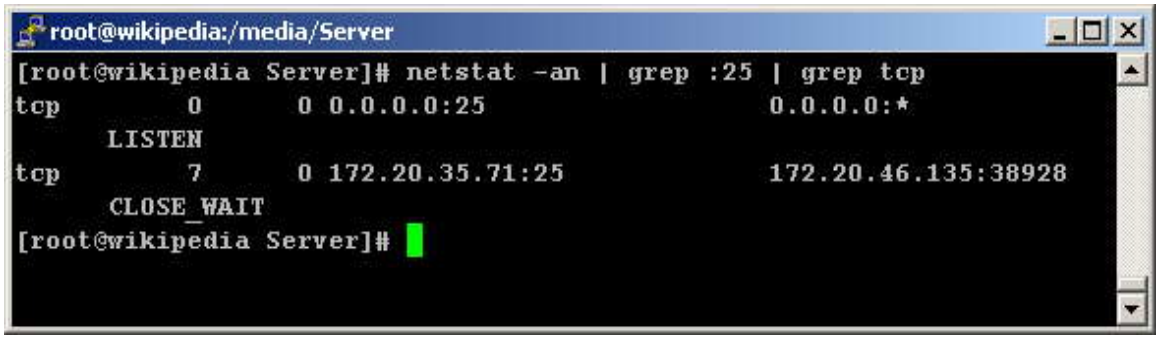
define(`SMART_HOST',`mail.my-site.com')



9. Converting From a Mail Client to a Mail Server

1) Determine which NICs sendmail is running on. You can see the interfaces on which sendmail is listening with the netstat command. Because sendmail listens on TCP port 25, you use net stat and grep for 25 to see a default configuration listening only on IP address 127.0.0.1 (loopback):

[root@linuxcoe]# netstat -an | grep :25 | grep tcp
tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN



[root@linuxcoe]#

2) Edit **sendmail.mc** to make sendmail listen on all interfaces. If sendmail is listening on the loopback interface only, you should comment out the daemon_options line in the **/etc/mail/sendmail.mc** file with **dnl** statements.

dnl
dnl This changes sendmail to only listen on the loopback
dnl device 127.0.0.1 and not on any other network
dnl devices. Comment this out if you want
dnl to accept email over the network.
dnl **DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')**
dnl

...
dnl
dnl We strongly recommend to comment this one out if you want
dnl to protect yourself from spam. However, the laptop and
dnl users on computers that do
dnl not have 24x7 DNS do need this.
dnl FEATURE(`accept_unresolvable_domains')dnl
dnl FEATURE(`relay_based_on_MX')dnl
dnl

3) Comment out the SMART_HOST Entry in sendmal.mc.

dnl define(`SMART_HOST',`mail.my-site.com')

4) Regenerate the sendmail.cf file, and restart sendmail. Again, you can do this with the restart script from the beginning of the chapter.

5) Make sure sendmail is listening on all interfaces (0.0.0.0).

[root@linuxcoe]# netstat -an | grep :25 | grep tcp
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN

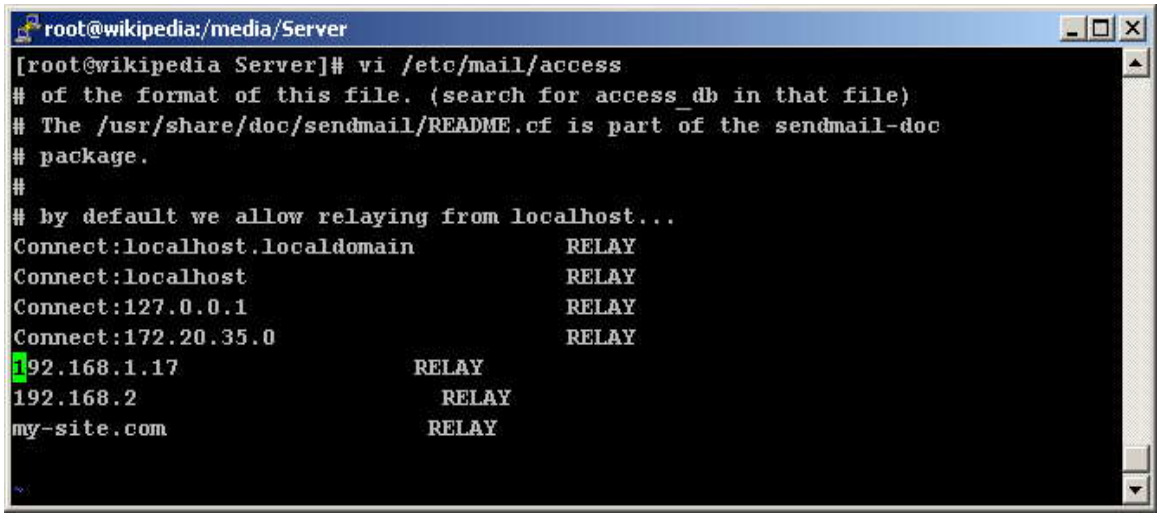
10. Configuration Files

The sendmail.mc File

dnl ***** Customised section 1 start *****
FEATURE(delay_checks)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_entire_domain)dnl
dnl
dnl
dnl ***** Customised section 1 end *****

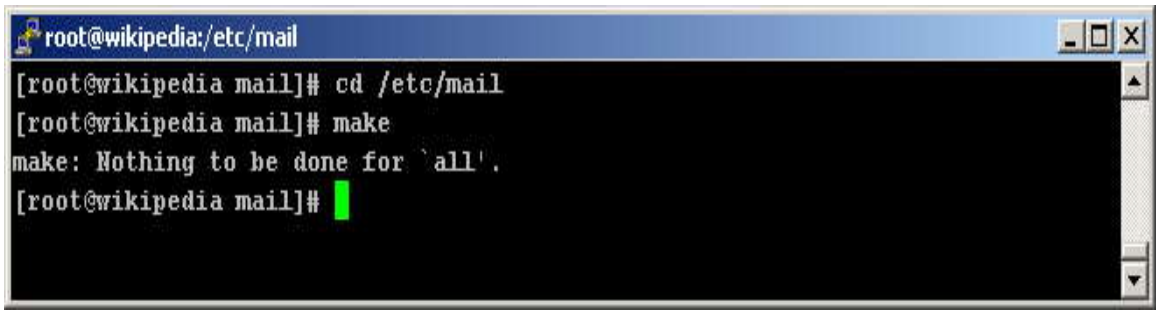
- **/etc/mail/access File**

The sample file that follows allows relaying for only the server itself (127.0.0.1, localhost), two client PCs on your home 192.168.1.X network, everyone on your 192.168.2.X network, and everyone passing e-mail through the mail server from servers belonging to my-site.com. Remember that a server will be considered a part of my-site.com only if its IP address can be found in a DNS reverse zone file:



You'll then have to convert this text file into a sendmail readable database file named **/etc/mail/access.db**. Here are the commands you need:

[root@linuxcoe]# cd /etc/mail
[root@linuxcoe]# make



Remember that the relay security features of this file may not work if you don't have a correctly configured /etc/hosts file.

- **/etc/mail/local-host-names File**

When sendmail receives mail, it needs a way of determining whether it is responsible for the mail it receives. It uses the **/etc/mail/local-host-names** file to do this. This file has a list of hostnames and domains for which sendmail accepts responsibility

- **/etc/mail/virtusertable file**

The **/etc/mail/virtusertable** file contains a set of simple instructions on what to do with received mail. The first column lists the target email address and the second column lists the local user's mail box, a remote email address, or a mailing list entry in the **/etc/aliases** file to which the email should be forwarded.

In this example, mail sent to:

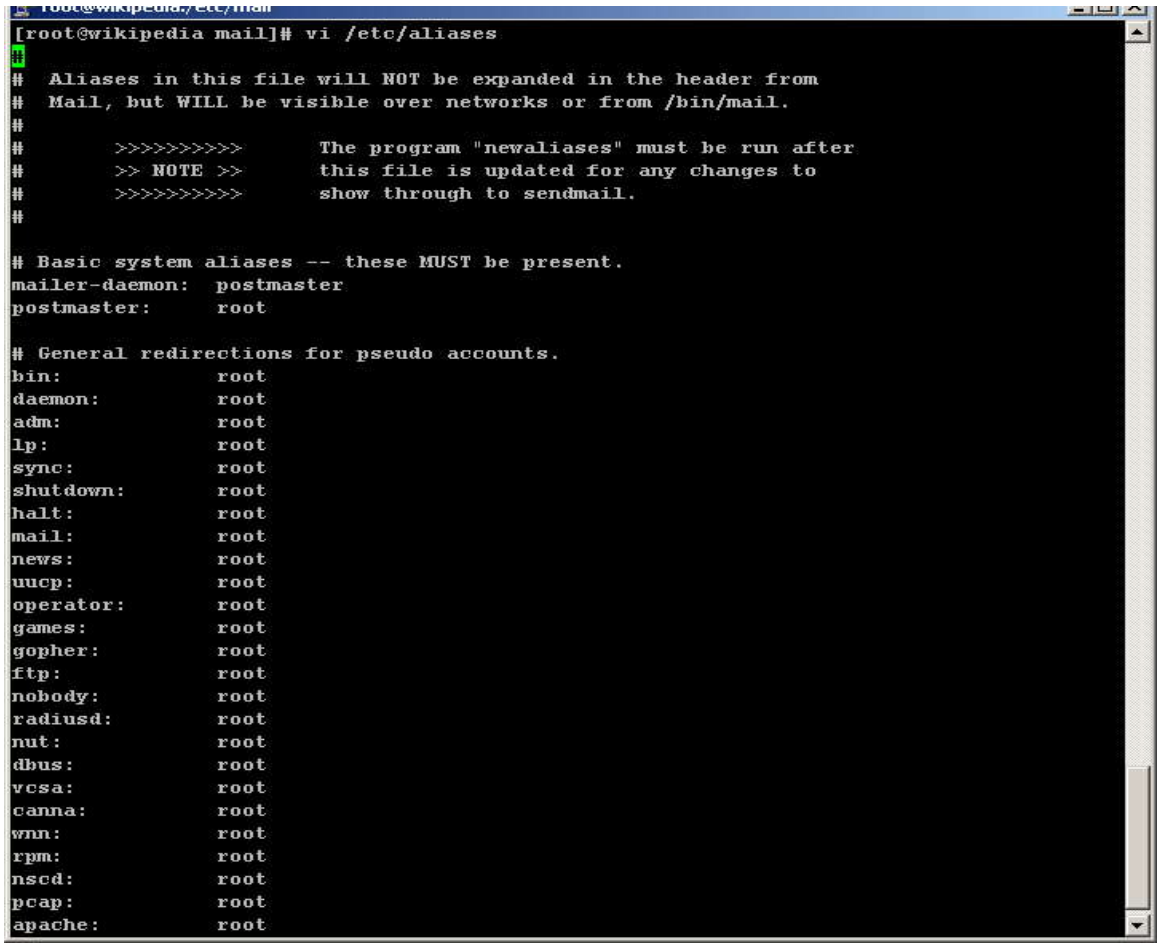
8. webmaster@another-site.com will go to local user (or mailing list) webmasters, all other mail to another-site.com will go to local user marc.
9. sales at my-site.com will go to the sales department at my-othersite.com.
10. paul and finance at my-site.com goes to local user (or mailing list) paul

After editing the /etc/mail/virtusertable file, you have to convert it into a sendmail-readable database file named /etc/mail/virtusertable.db with two commands:

[root@linuxcoe]# cd /etc/mail
[root@linuxcoe]# make

- **The /etc/aliases File**

/etc/aliases file as a mailing list file. The first column has the mailing list name and the second column has the members of the mailing list .



[root@linuxcoe]# **newaliases**

- Mail to "directors@my-site.com" goes to users "peter", "paul" and "mary".
- # Directors of my SOHO company
- directors: peter,paul,mary
- Mail sent to "family@my-site.com" goes to users "grandma", "brother" and "sister"
- # My family
- family: grandma,brother,sister
- Mail sent to admin-list gets sent to all the users listed in the file /home/mailings/admin-list.
- # My mailing list file
- admin-list: ":include:/home/mailings/admin-list"

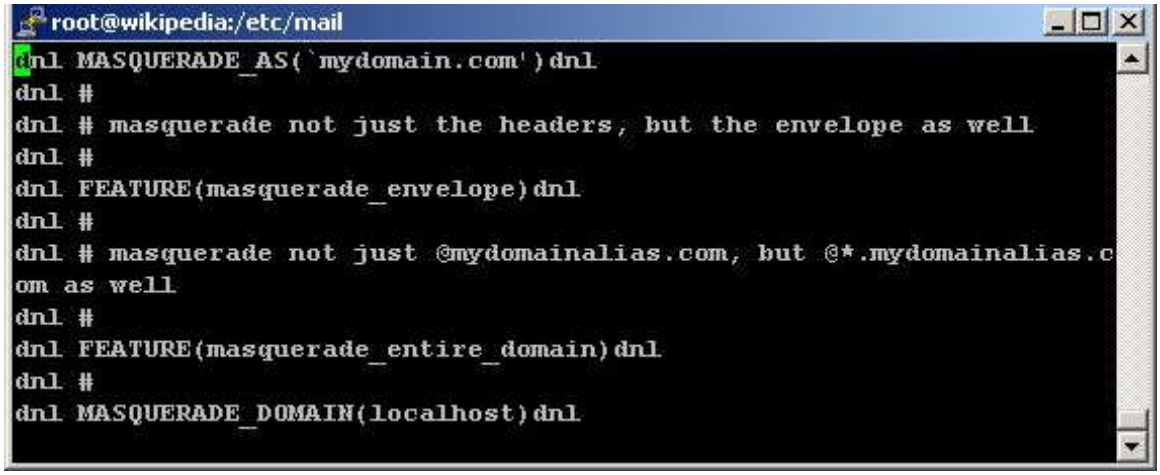
11. Sendmail Masquerading

- Configure your email client, such as Outlook Express, to set your email address to user@mysite.com. (I'll explain this in the "Configuring Your POP Mail Server" section.).
- Set up masquerading to modify the domain name of all traffic originating from and passing trough your mail server

Configuring masquerading

By editing your sendmail.mc configuration file and adding s ome masquerading commands and directives:

FEATURE(always_add_domain)dnl
FEATURE(`masquerade_entire_domain')dnl
FEATURE(`masquerade_envelope')dnl
FEATURE(`allmasquerade')dnl
MASQUERADE_AS(`my-site.com')dnl
MASQUERADE_DOMAIN(`my-site.com.')dnl
MASQUERADE_DOMAIN(localhost)dnl
MASQUERADE_DOMAIN(localhost.localdomain)dnl



Testing Masquerading

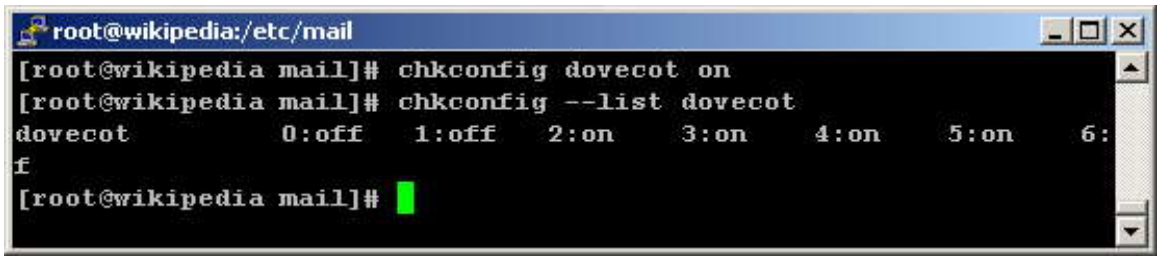
The best way of testing masquerading from the Linux command line is to use the "mail -v username" command. I have noticed that "sendmail -v username" ignores masquerading altogether. You should also tail the /var/log/maillog file to verify that the masquerading is operating correctly and check the envelope and header of test email received by test email accounts.

12. Installing POP Mail Server

- rpm –ivh dovecot-1.0-1.2.rc15.el5.i386.rpm

Starting POP Mail Server

[root@linuxcoe]# chkconfig dovecot on



To start, stop, and restart dovecot after booting, use

[root@linuxcoe]# service dovecot restart

Restart the dovecot process every time you make a change to the configuration files for the changes to take effect on the running process. You can also test whether the dovecot process is running with the pgrep command:

[root@linuxcoe]# pgrep dovecot

You should get a response of plain old process ID numbers:

The /etc/dovecot.conf File

By default dovecot will act as a server for IMAP, secure IMAP (IMAPS), POP and secure POP (POPS). You can limit this list by editing the protocols line in the /etc/dovecot.conf file and then restarting dovecot for the change to take effect. In the example below dovecot is configured to serve only POP3.

```
#
# File /etc/dovecot.conf sample
 #

 # Protocols we want to be serving imap imaps pop3 pop3s
 #protocols = imap imaps pop3 pop3s
  protocols = pop3
```

You can then use the **netstat** command to do a simple preliminary test to make sure dovecot is serving POP3 only.

```
[root@linuxcoe]# netstat -a | egrep -i 'pop|imap'
 tcp        0      0 *:pop3                  *:*                     LISTEN
[root@linuxcoe]#
```

13. Configure Your Windows Mail Programs

All your POP e-mail accounts are really only regular Linux user accounts in which sendmail has deposited mail. You can now configure your e-mail client such as Outlook Express to use your use your new POP/SMTP mail server quite easily. To configure POP Mail, set your POP mail server to be the IP address of your Linux mail server. Use your Linux user username and password when prompted.

Next, set your SMTP mail server to be the IP address/domain name of your Linux mail server.

14. Testing Sendmail

1) Telnet to the mail server on port 25. You should get a response with a 220 status code.

```
[root@linuxcoe]# telnet mail.my-site.com 25
Trying mail.my-site.com...
Connected to mail.my-site.com.
Escape character is '^]'.
220 mail.my-site.com ESMTP server ready
```

2) Use the hello command to tell the mail server the domain you belong to. You should receive a message with a successful status 250 code at the beginning of the response.

```
helo another-web-site.org
```

250 mail.my-site.com Hello c-24-4-97-110.client.comcast.net [24.4.97.110], pleased to meet you.

3) Inform the mail server from which the test message is coming with the MAIL FROM: statement.

MAIL FROM:sender@another-web-site.org
250 2.1.0 sender@another-web-site.org... Sender ok

4) Tell the mail server to whom the test message is going with the " RCPT TO:" statement.

RCPT TO: user@my-site.com
250 2.1.5 user@my-site.com... Recipient ok

5) Prepare the mail server to receive data with the DATA statement

DATA
354 Enter mail, end with "." on a line by itself

6) Type the string "subject:" then type a subject. Type in your text message, ending it with a single period on the last line. For example.

Subject: Test Message
Testing sendmail interactively
.
250 2.0.0 iA75r9si017840 Message accepted for delivery

7) Use the QUIT command to end the session.

QUIT
221 2.0.0 mail.my-site.com closing connection
Connection closed by foreign host.
[root@linuxcoe]#

- **/var/log/maillog File**

Sendmail writes all its status messages in the **/var/log/maillog** file, always monitor this file whenever you are doing changes. Open two TELNET, SSH, or console windows. Work in one of them and monitor the sendmail status output in the other using the command

[root@linuxcoe]# tail -f /var/log/maillog

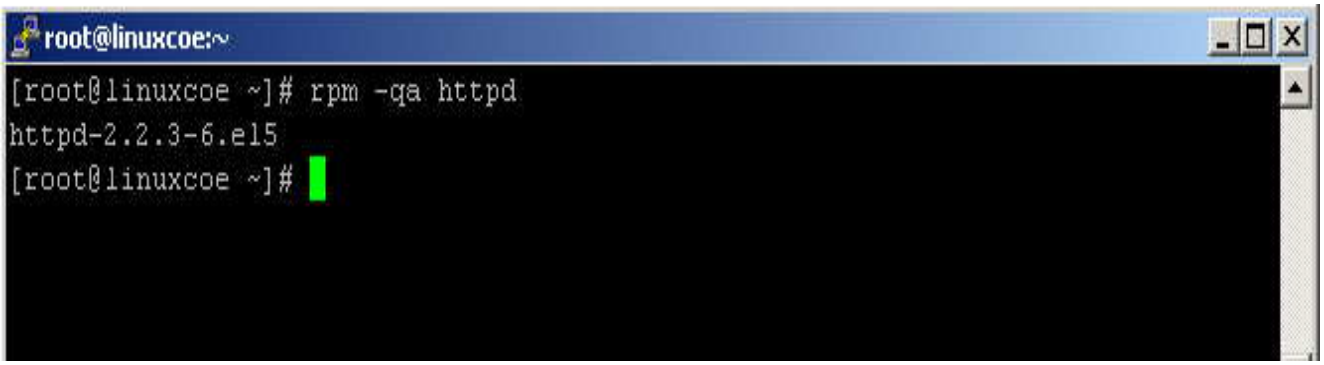
21. HTTP / Apache

Hypertext Transfer Protocol is the set of rules for transferring files on the World Wide Web

1. Packages

httpd-2.2.3-6.el5.i386.rpm
httpd-devel-2.2.3-6.el5.i386.rpm

Check out whether HTTPD service is installed.



2. Configuration File

The primary file for configuring your Apache Web server is **httpd.conf** (located in the /etc/httpd/conf directory).

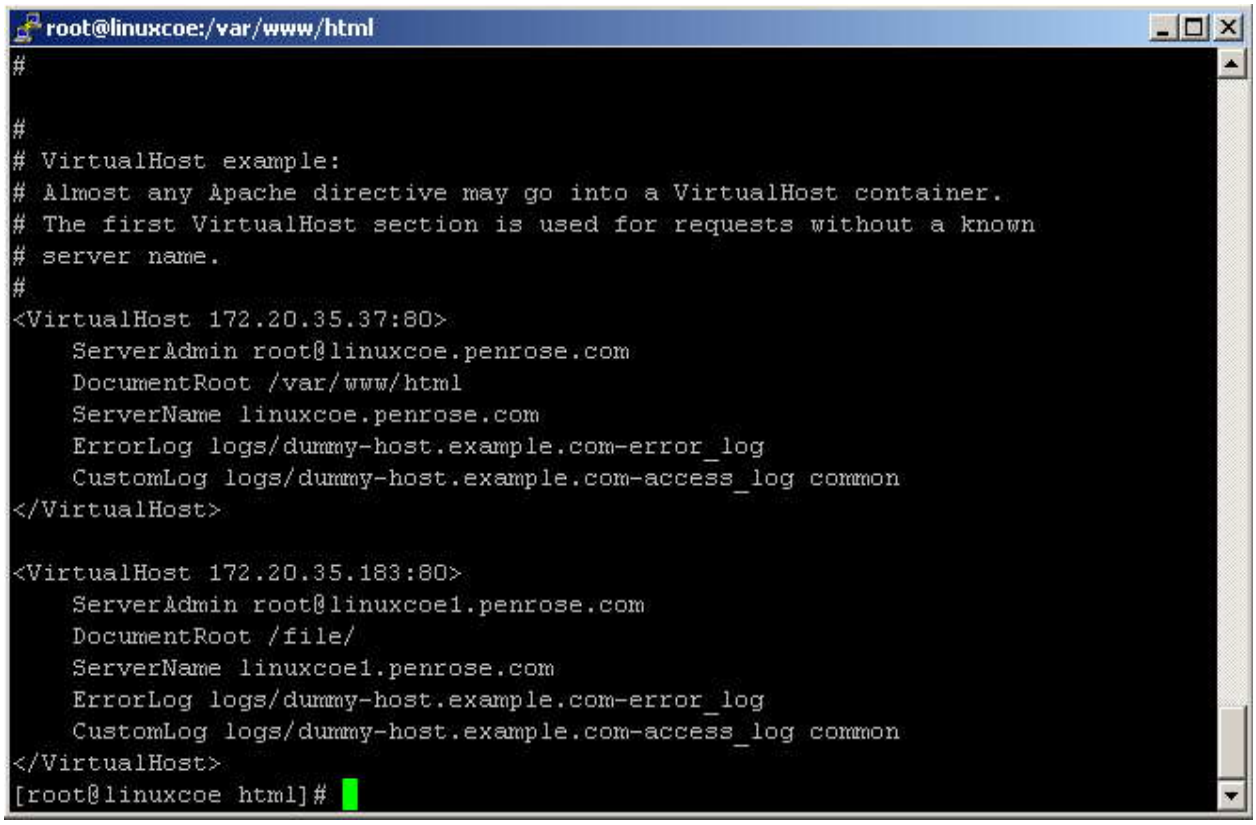
By default, all web page files will be in **/var/www/html/** directory .A file named index.html is created with html code for the web page. This is the main configuration page and an entry is made in the DNS forward zone file.

3. Named Virtual Hosting

You can make your Web server host more than one site per IP address by using Apache's named virtual hosting feature. You use the NameVirtualHost directive in the /etc/httpd/conf/httpd.conf file to tell Apache which IP addresses will participate in this feature.

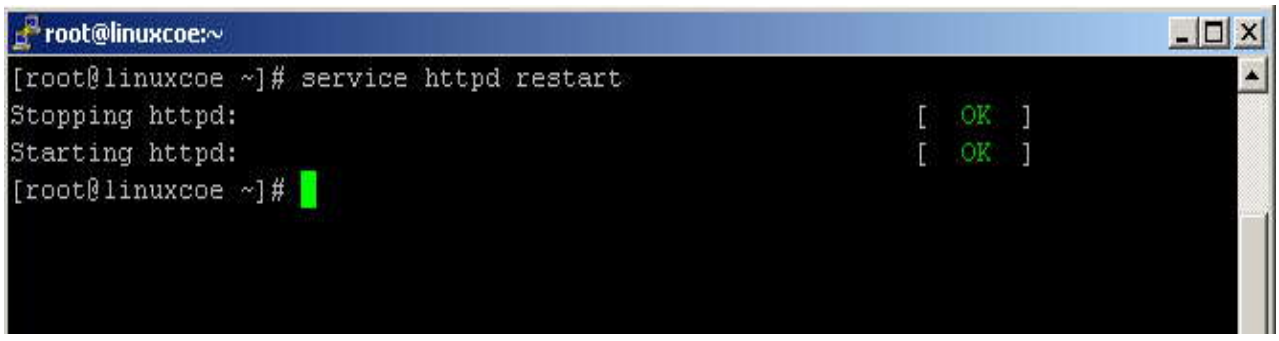
The <VirtualHost> containers in the file then tell Apache where it should look for the Web pages used on each Web site. You must specify the IP address for which each <VirtualHost> container applies.

So in the configuration file httpd.conf, add the following in Virtual Host as shown:

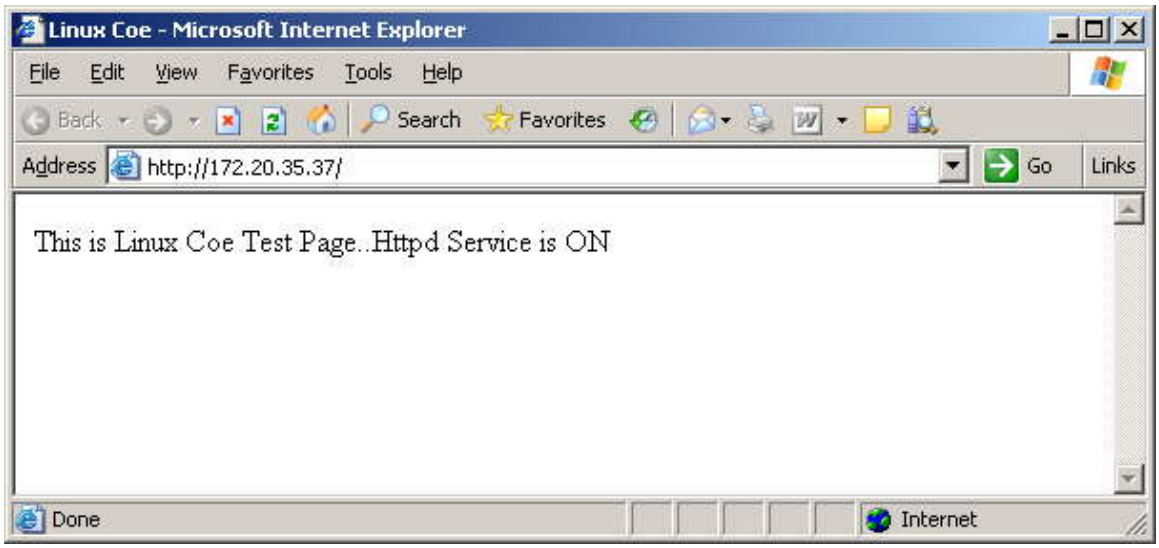


Here 172.20.35.37 and 172.20.35.183 are the two IP assigned to NIC, where 172.20.35.183 is virtual IP. Keep both the index files in different path as shown above. Here one file is kept in /var/www/html and the other in /file.

Then restart the httpd service once as shown below:



Then from the web browser, in other system, type the corresponding hostnames or i p address (both the ip addresses) of the http server and check out whether web page is displaying.



In /etc/hosts file, enter the corresponding IPs with the hostname in Linux server. Go to other windows system and add the corresponding hostname in the hosts file, otherwise type the ip in the browser, so that we can get the corresponding web page there.

22. Configuring Linux as a Router

By default any modern Linux distributions will have **IP Forwarding disabled**. This is normally a good idea, as most peoples will not need IP Forwarding, but if we are setting up a **Linux router/gateway** or maybe a **VPN server** (pptp or ipsec) or just a plain **dial-in server** then we will need to **enable forwarding**. This can be done in several ways that I will present bellow.

a. Check if IP Forwarding is enabled

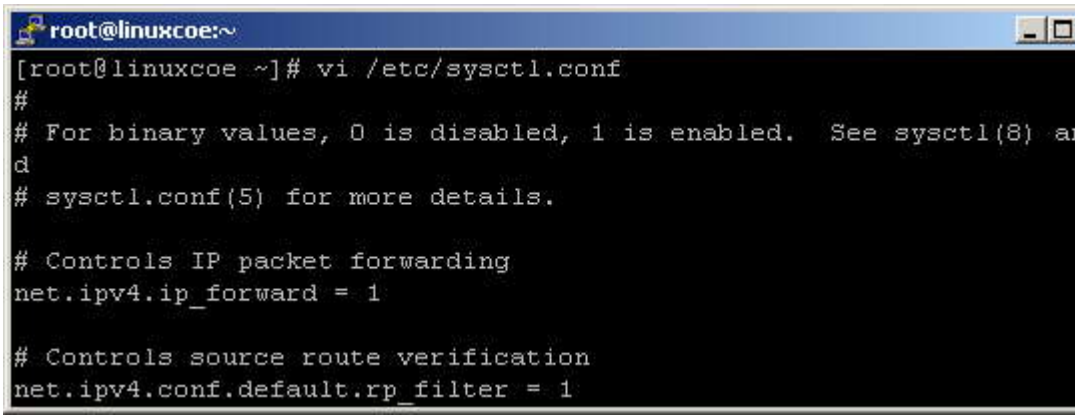
We have to query the **sysctl kernel** value **net.ipv4.ip_forward** to see if forwarding is enabled or not:

Using sysctl:

b. Enable IP Forwarding on the fly

c. Permanent setting using /etc/sysctl.conf

i.



d. To enable the changes made in sysctl.conf you will need to run the command

```
#sysctl -p /etc/sysctl.conf

            or
#service network restart
```

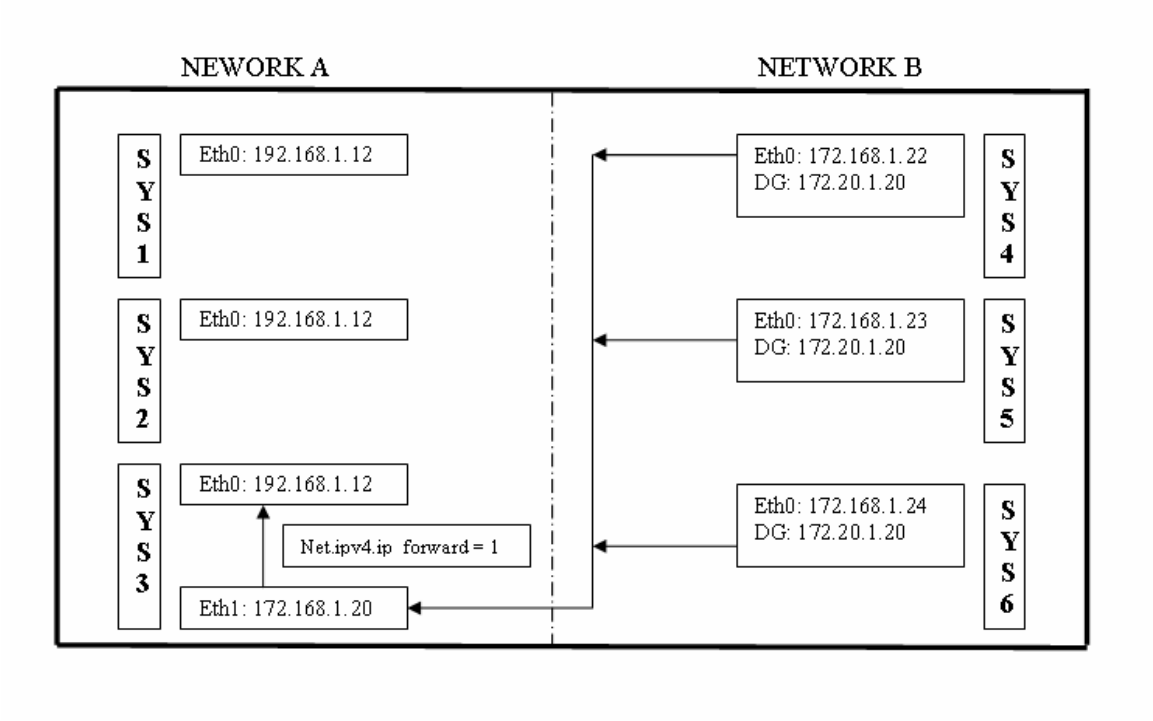
Scenario:

Now let us assume that there are 6 computers connected together but are on 2 different networks

Network A
Sys1-192.168.1.12
Sys2-192.168.1.13
Sys3-192.168.1.14

Network B
Sys4-172.168.1.22
Sys5-172.168.1.22
Sys6-172.168.1.22

If any system tries pinging to another in a different network it would result in "Network not reachable" error.



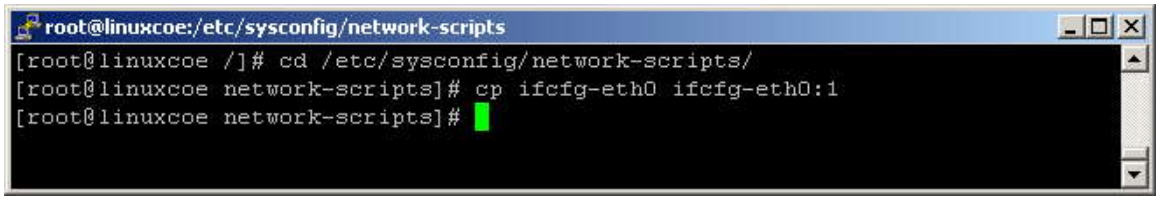
For a system in Network B to communicate to a system in Network A, it needs a default gateway in its IP range.

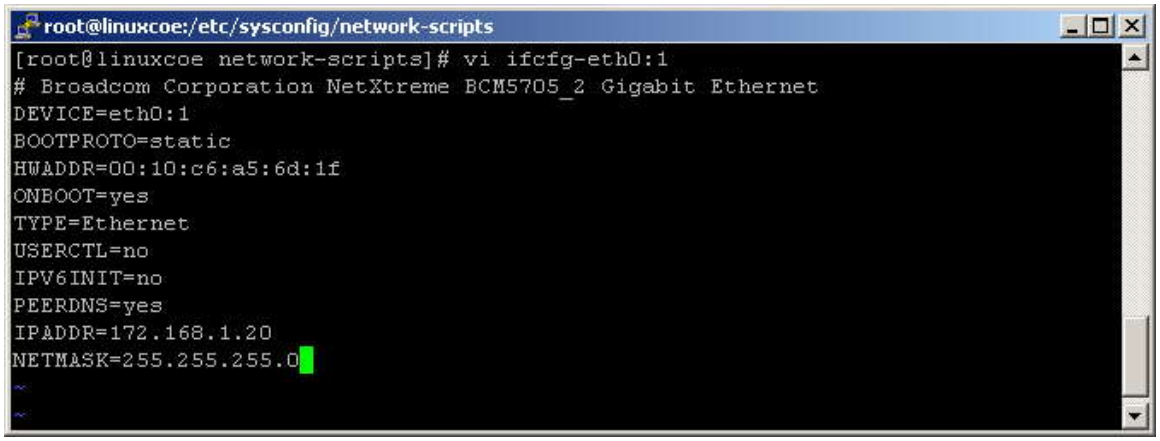
As shown above the sys3 has IP forwarding enabled and also has a 2nd IP address 172.20.1.20.

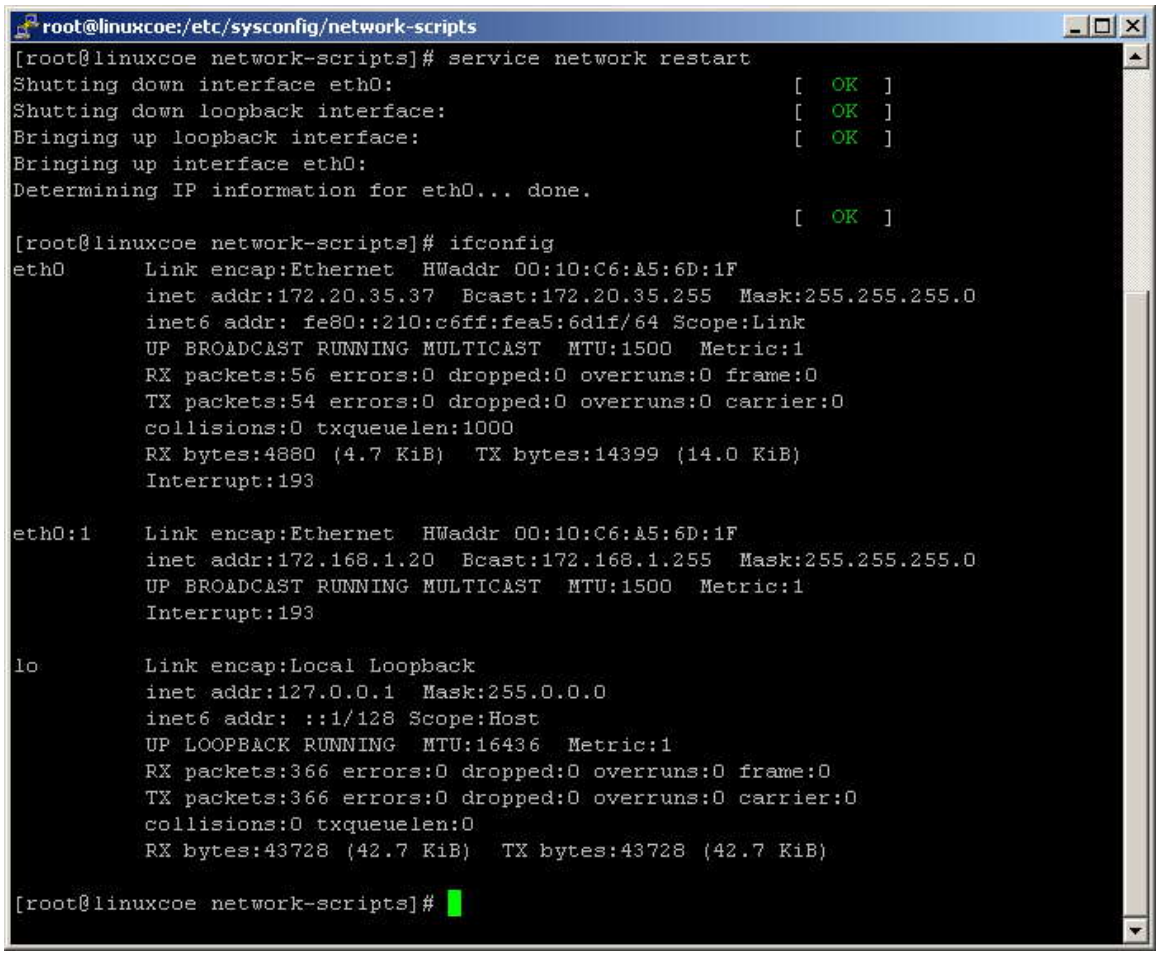
This can be done in two ways.

Add a 2nd Ethernet Adapter

Add a device Alias







Once the sefault gateway is specified in /etc/sysconfig/network/ifcfg-eth0 on the computers in Network B they will send packets to the router sys3 which will forward the packets to required systems on network A.

---

23. Squid – Proxy Server

What is squid

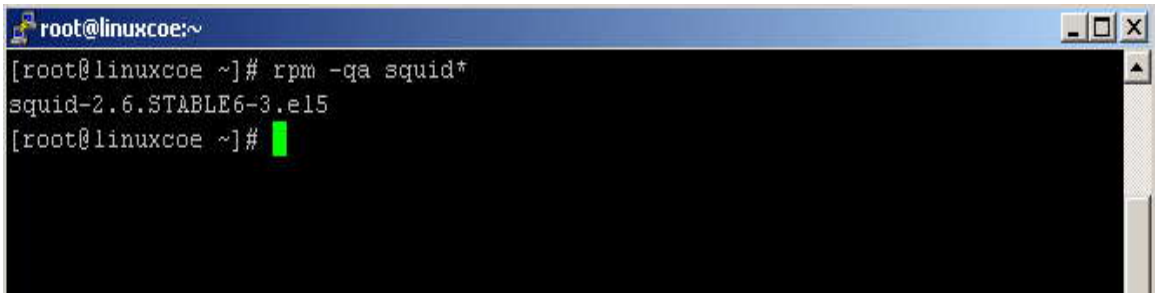
SQUID is a program that caches Web and other Internet content in a Unix-based proxy server closer to the user than the content-originating site

1. **Daemons**

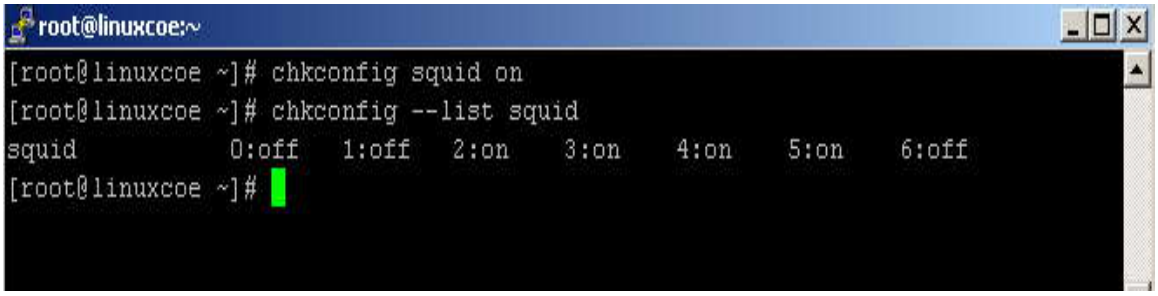
squid - Runs the squid proxy web server.

2. Check out whether squid is installed

chkconfig squid on



output

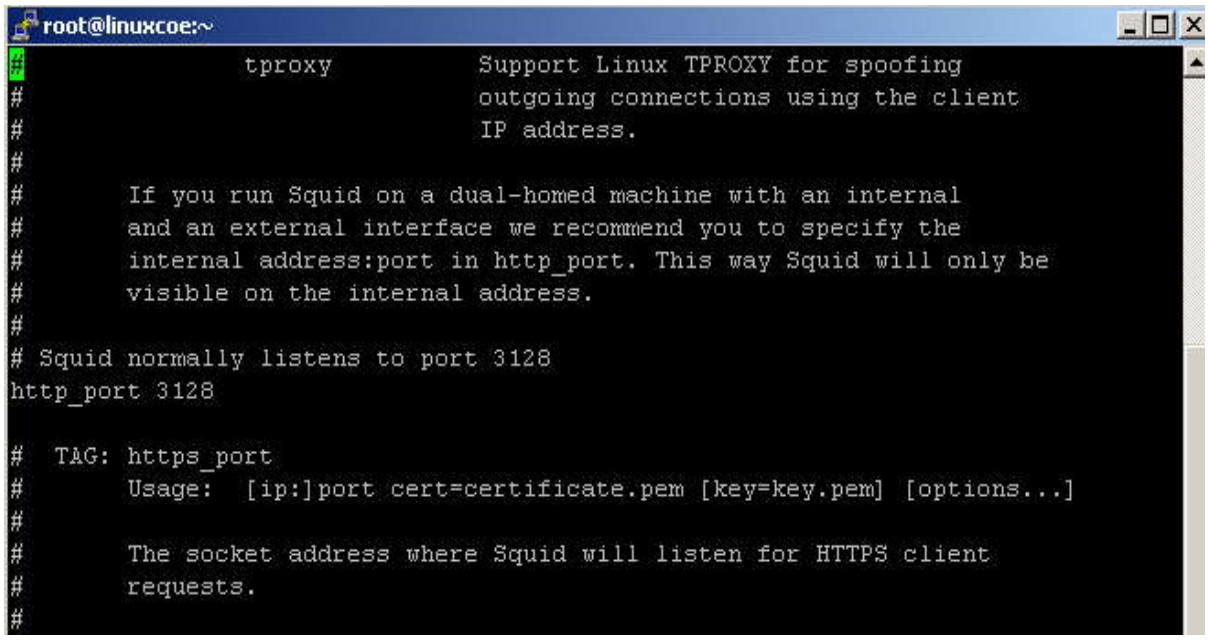


3. Visible Host Name

Squid will fail to start if you don't give your server a hostname. You can set this with the visible_hostname parameter. Here, the hostname is set to the real name of the server linuxcoe

visible_hostname linuxcoe

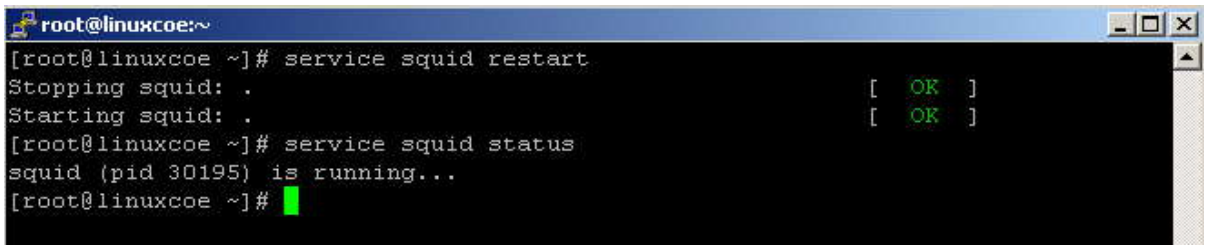
vi /etc/squid/squid.conf



Enable http_port 3128. In the above screenshot http_port is enabled.

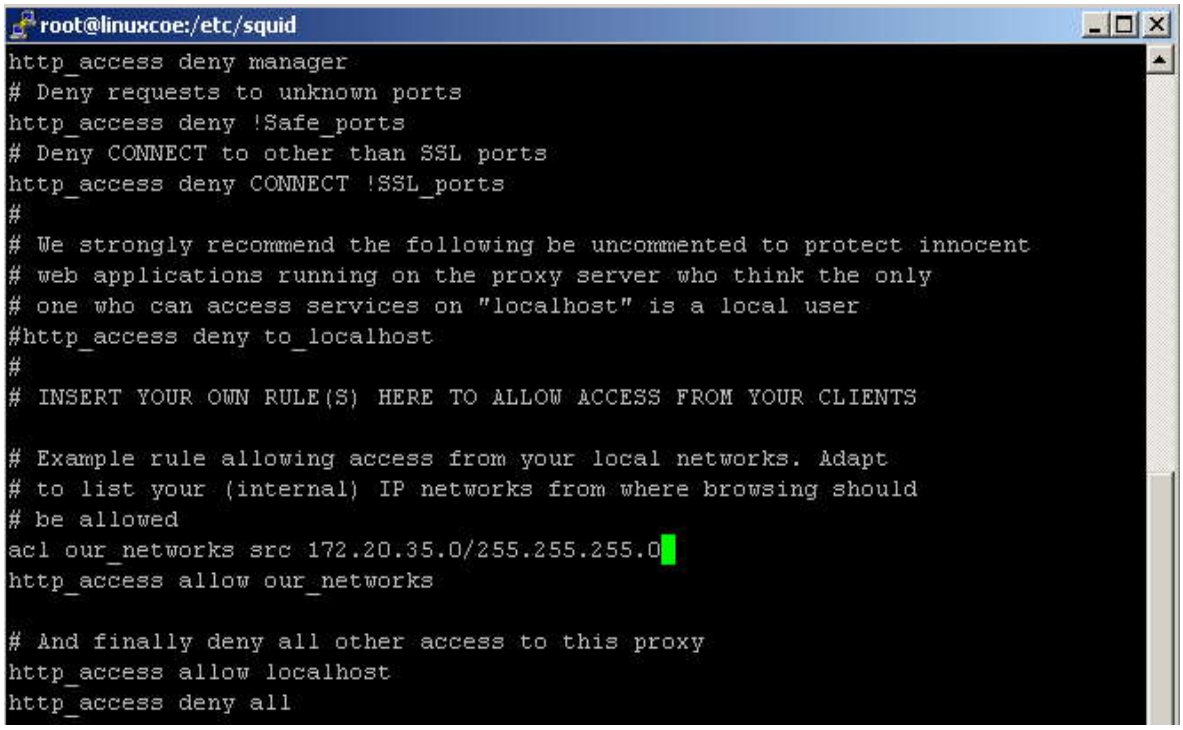
Save and exit

Start the squid service



4. Insert your rule to access from clients

acl our_networks src 172.20.35.0/255.255.255.0
http_access allow our_networks

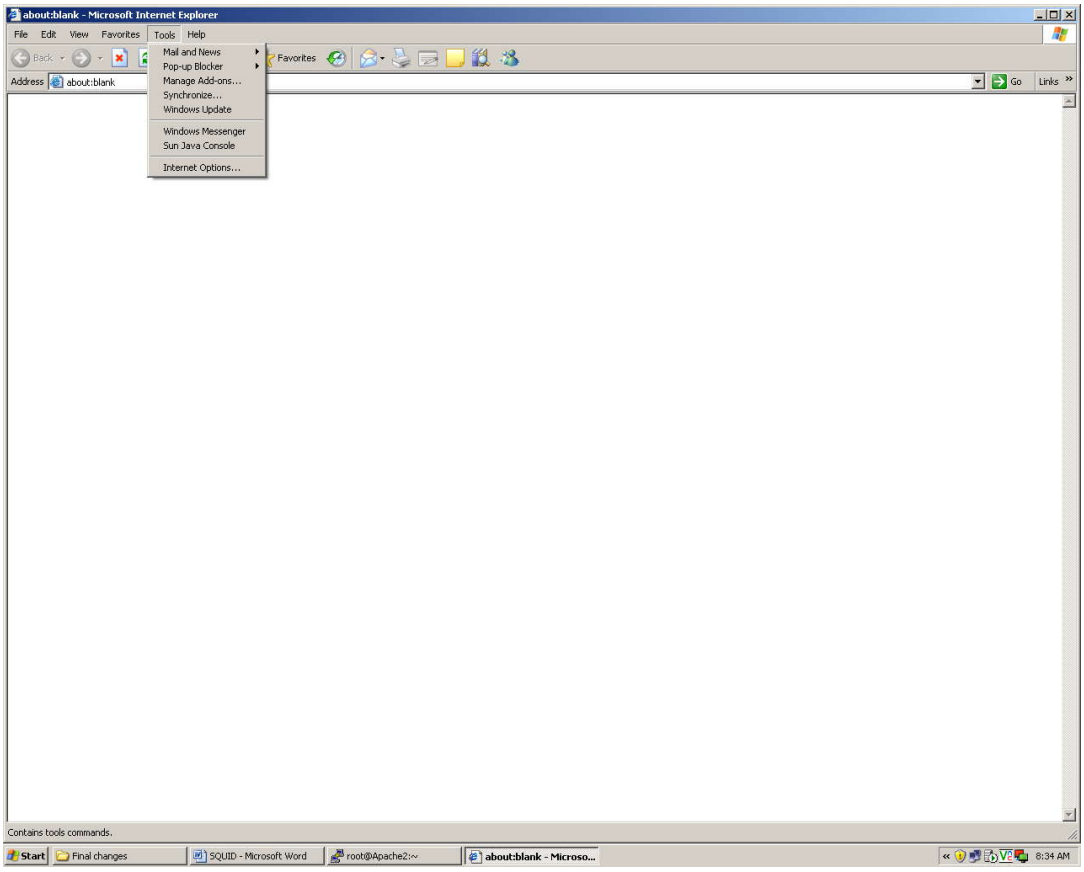


Squid needs to be restarted for changes to the configuration file can take effect.

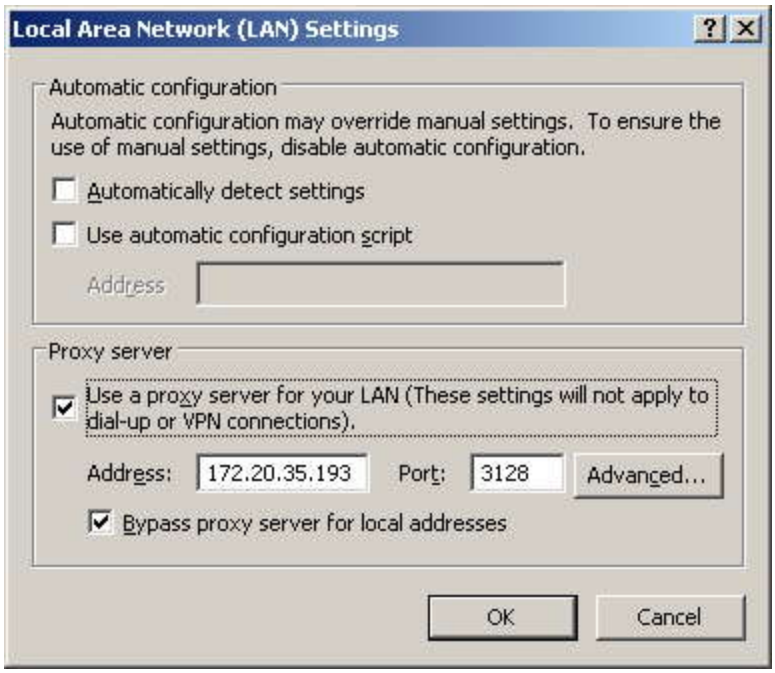
5. Testing squid in Client machine

For example, to make these changes using Internet Explorer

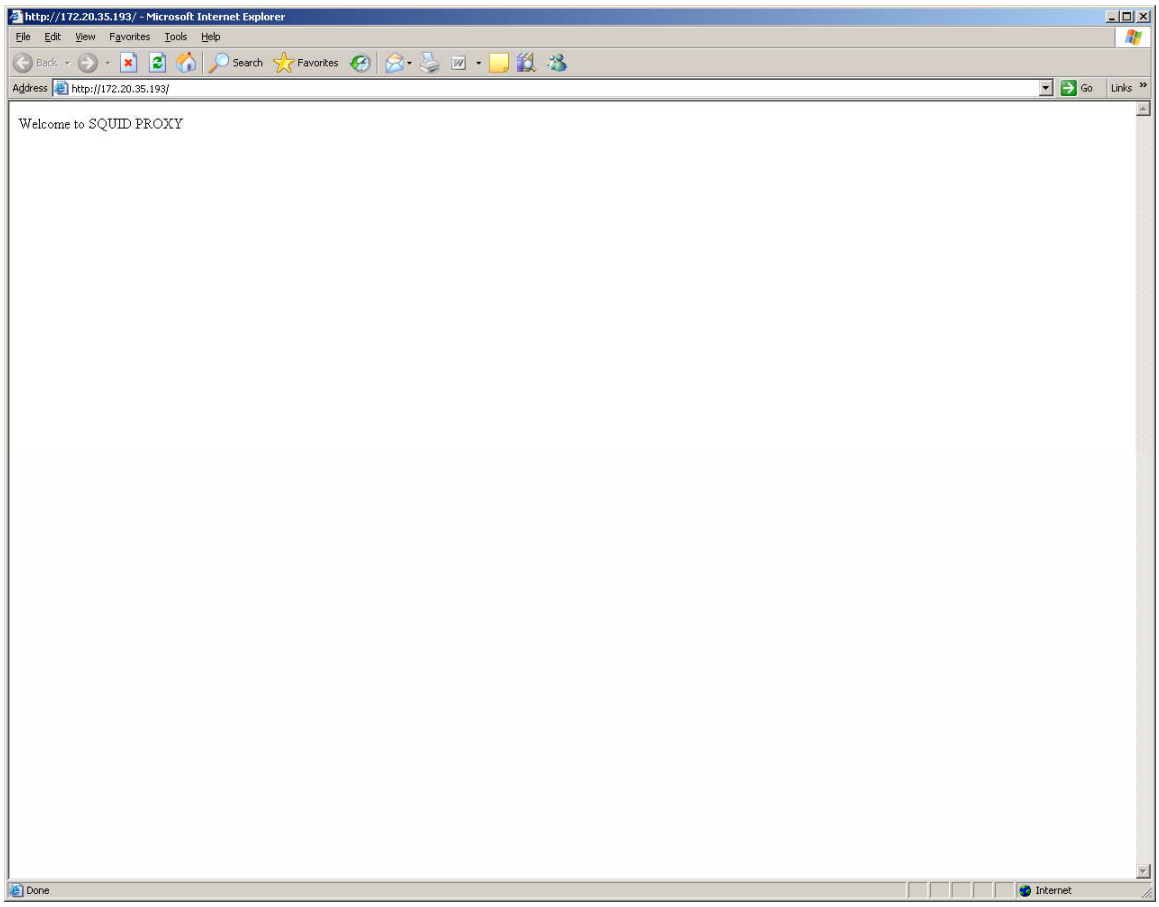
1. Click on the "Tools" item on the menu bar of the browser.
2. Click on "Internet Options"
3. Click on "Connections"
4. Click on "LAN Settings"
5. Configure with the address and TCP port (3128 default) used by your Squid server.



.



Edit index.html in /var/www/html/ and restart httpd service. Open browser Check for proxy ipaddress



6. Access Control Lists

You can limit users ability to browse the Internet with access control lists (ACLs). Each ACL line defines a particular type of activity, such as an access time or source network, they are then linked to an http_access statement that tells Squid whether or not to deny or allow traffic that matches the ACL

Restricting Web Access by Time

create access control lists with time parameters. For example, you can allow only business hour access from the home network, while always restricting access to host 192.168.1.23.

```
#
# Add this to the bottom of the ACL section of squid.conf
#
acl home_network src 172.20.35.0/255.255.255.0
acl business_hours time M T W H F 9:00-17:00
acl RestrictedHost src 192.168.1.23
```

```
#
# Add this at the top of the http_access section of squid.conf
#
http_access deny RestrictedHost
http_access allow home_network business_hours
```

Or, you can allow morning access only:

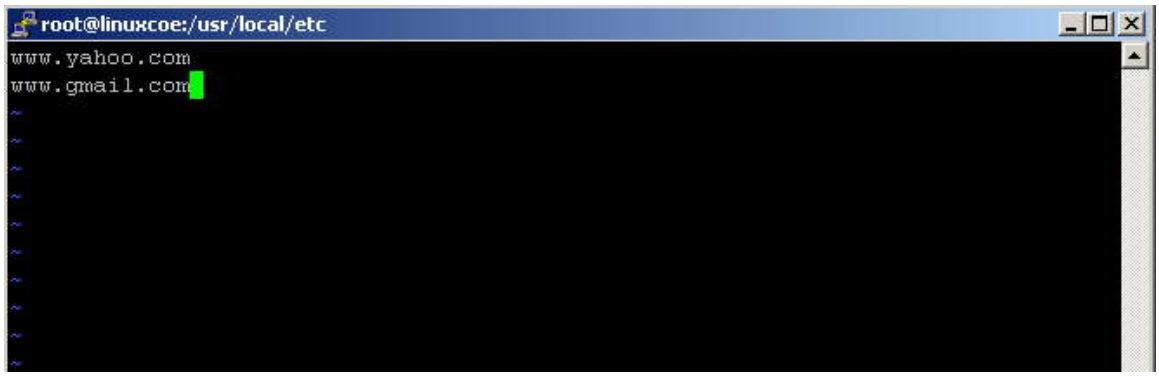
```
#
# Add this to the bottom of the ACL section of squid.conf
#
acl mornings time 08:00-12:00

#
# Add this at the top of the http_access section of squid.conf
#
http_access allow mornings
```

Restricting Access to specific Web sites

Squid is also capable of reading files containing lists of web sites and/or domains for use in ACLs. In this example we create to lists in files named /usr/local/etc/allowed-sites.squid and /usr/local/etc/restricted-sites.squid

File: /usr/local/etc/allowed-sites.squid

File: /usr/local/etc/restricted-sites.squid



These can then be used to always block the restricted sites and permit the allowed sites during working hours

```
#
# Add this to the bottom of the ACL section of squid.conf
#
acl home_network src 172.20.35.0/24
acl business_hours time M T W H F 9:00-17:00
acl GoodSites dstdomain "/usr/local/etc/allowed-sites.squid"
acl BadSites  dstdomain "/usr/local/etc/restricted-sites.squid"

#
# Add this at the top of the http_access section of squid.conf
#
http_access deny BadSites
http_access allow home_network business_hours GoodSites
```

Restricting Web Access By IP Address

You can create an access control list that restricts Web access to users on certain networks. In this case, it's an ACL that defines a home network of 172.20.35.0

```
#
# Add this to the bottom of the ACL section of squid.conf
#
acl home_network src 192.168.1.0/255.255.255.0
```

You also have to add a corresponding http_access statement that allows traffic that matches the ACL:

```
#
# Add this at the top of the http_access section of squid.conf
#
http_access allow home_network
```

7. Squid Transparent Proxy Configuration

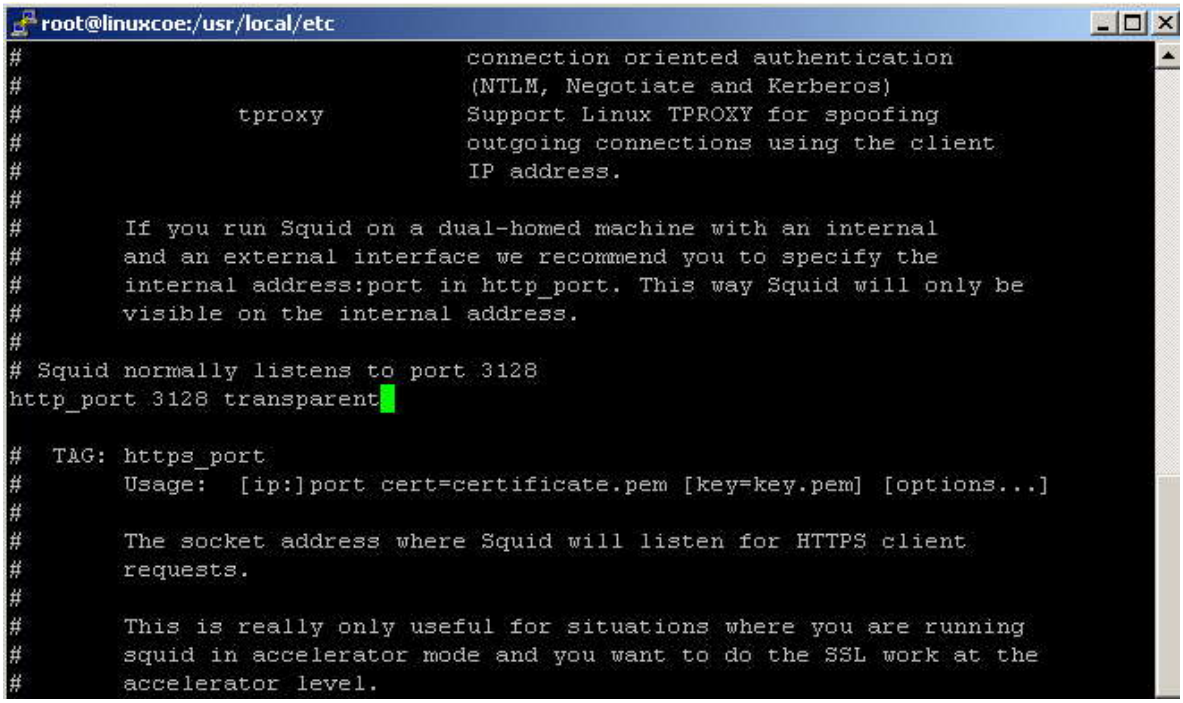
Your first step will be to modify your squid.conf to create a transparent proxy. The procedure is different depending on your version of Squid.

In older versions of Squid, transparent proxy was achieved through the use of the httpd_accel options which were originally developed for http acceleration. In these cases, the configuration syntax would be as follows:

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

In version 2.6 and beyond Squid simply require you to add the word "transparent" to the default "http_port 3128" statement. In this example, Squid not only listens on TCP port 3128 for proxy connections, but will also do so in transparent mode.

```
http_port 3128 transparent
```


root@linuxcoe:/usr/local/etc
connection oriented authentication
(NTLM, Negotiate and Kerberos)
tproxy Support Linux TPROXY for spoofing
outgoing connections using the client
IP address.
#
If you run Squid on a dual-homed machine with an internal
and an external interface we recommend you to specify the
internal address:port in http_port. This way Squid will only be
visible on the internal address.
#
Squid normally listens to port 3128
http_port 3128 transparent
#
TAG: https_port
Usage: [ip:]port cert=certificate.pem [key=key.pem] [options...]
#
The socket address where Squid will listen for HTTPS client
requests.
#
This is really only useful for situations where you are running
squid in accelerator mode and you want to do the SSL work at the
accelerator level.

24. IPTABLES

Iptables is a generic table structure for the definition of rulesets .Each rule within an IP table consists of a number of classifiers (iptables matches) and one connected action (iptables target)

INSTALLATION

iptables-1.2.9-1.0.i386.rpm

1. How to start /stop / restart iptables

You can start, stop, and restart iptables after booting by using the commands:

service iptables start

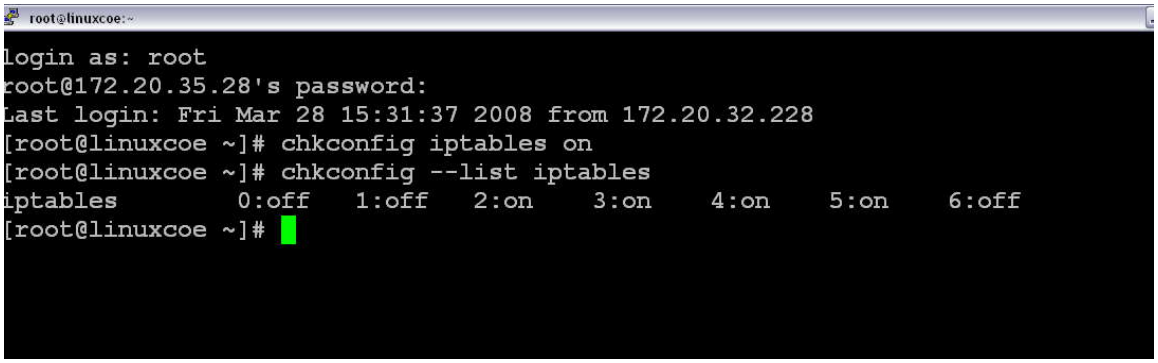
service iptables stop

service iptables restart

2. To get iptables configured to start at boot time,

chkconfig iptables on

Output:



There are three tables in total. The first is the mangle table which is responsible for the alteration of quality of service bits in the TCP header. This is hardly used in a home or SOHO environment.

The second table is the filter queue which is responsible for packet filtering. It has three built-in chains in which you can place your firewall policy rules. These are the:

- Forward chain: Filters packets to servers protected by the firewall.
- Input chain: Filters packets destined for the firewall.
- Output chain: Filters packets originating from the firewall.

The third table is the nat queue which is responsible for network address translation. It has two built-in chains; these are:

- Pre-routing chain: NATs packets when the destination address of the packet needs to be changed.
- Post-routing chain: NATs packets when the source address of the packet needs to be changed

3. Determining the Status of iptables

You can determine whether iptables is running or not via the service iptables status command. Fedora Core will give a simple status message. For example

service iptables status

4. Processing For Packets Routed By the Firewall

Queue Type	Queue Function	Packet Transformation Chain in queue	Chain Function
Filter	Packet filtering	FORWARD	Filters packets to servers accessible by another NIC on the firewall.
		INPUT	Filters packets destined to the firewall.
		OUTPUT	Filters packets

			originating from the firewall
Nat	Network Address Translation	PREROUTING	Address translation occurs before routing. Facilitates the transformation of the destination IP address to be compatible with the firewall's routing table. Used with NAT of the destination IP address, also known as **destination NAT** or **DNAT**.
		POSTROUTING	Address translation occurs after routing. This implies that there was no need to modify the destination IP address of the packet as in pre-routing. Used with NAT of the source IP address using either one-to-one or many-to-one NAT. This is known as **source NAT**, or **SNAT**.
Mangle	TCP header modification	OUTPUT PREROUTING POSTROUTING	Network address translation for packets generated by the firewall. (Rarely used in SOHO environments) Modification of the TCP packet quality

		OUTPUT INPUT FORWARD	of service bits before routing occurs(Rarely used in SOHO environments)

You need to specify the table and the chain for each firewall rule you create. There is an exception: Most rules are related to filtering, so iptables assumes that any chain that's defined without an associated table will be a part of the filter table.

5. TARGETS

Target	Description	Most Common Options
ACCEPT	Iptables stops further processing. The packet is handed over to the end application or the operating system for processing	N/A
DROP	iptables stops further processing. The packet is blocked	N/A

REJECT	Works like the DROP target, but will also return an error message to the host sending the packet that the packet was blocked	--reject-with *qualifier* The qualifier tells what type of reject message is returned. Qualifiers include: icmp-port-unreachable (default) icmp-net-unreachable icmp-host-unreachable icmp-proto-unreachable icmp-net-prohibited icmp-host-prohibited tcp-reset echo-reply
MASQUERADE	Used to do Source Network Address Translation. By default the source IP address is the same as that used by the firewall's interface	*[--to-ports <port>[-<port>]]* Specifies the range of source ports to which the original source port can be mapped.

5. Iptables Command

Iptables command Switch	Description

-t <table>	If you don't specify a table, then the filter table is assumed. As discussed before, the possible built-in tables include: filter, nat, mangle
-j <target>	Jump to the specified target chain when the packet matches the current rule.
-A	Append rule to end of a chain
-F	Flush. Deletes all the rules in the selected table
-p <protocol-type>	Match protocol. Types include, icmp, tcp, udp, and all
-s <ip-address>	Match source IP address
-d <ip-address>	Match destination IP address
-i <interface-name> -o <interface-name>	Match "input" interface on which the packet enters. Match "output" interface on which the packet exits

Scenario 1

In this scenario, iptables is being configured to allow the firewall to accept TCP packets coming in on interface eth0 from any IP address destined for the firewall's IP address of 172.20.35.28. The 0/0 representation of an IP address means any.

iptables -A INPUT -s 0/0 -i eth0 -d 172.10.35.28 -p TCP -j ACCEPT

Output:

```
root@linuxcoe:~
[root@linuxcoe ~]# iptables -A INPUT -s 0/0 -i eth0 -d 172.20.35.28 -p TCP -j
CEPT
[root@linuxcoe ~]# service iptables save
Saving firewall rules to /etc/sysconfig/iptables:          [  OK  ]
[root@linuxcoe ~]#
```

Scenario 2

Common ICMP (Ping)

Matches used with --icmp-type	Description
--icmp-type <type>	The most commonly used types are echo-reply and echo-request

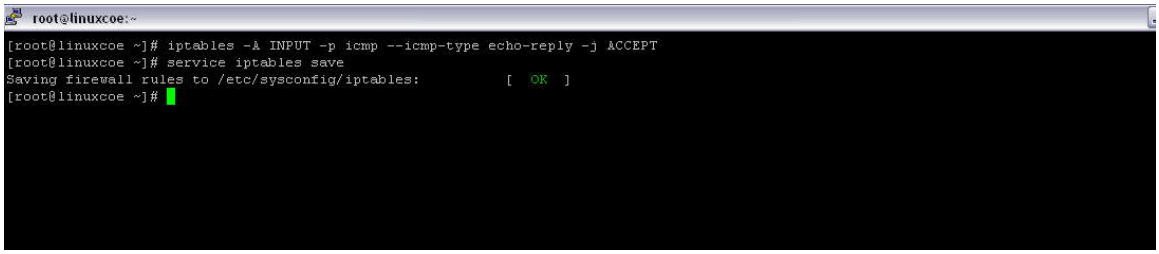
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT

Output :

```
root@linuxcoe:~
[root@linuxcoe ~]# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
[root@linuxcoe ~]# service iptables save
Saving firewall rules to /etc/sysconfig/iptables:          [  OK  ]
[root@linuxcoe ~]#
```

iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

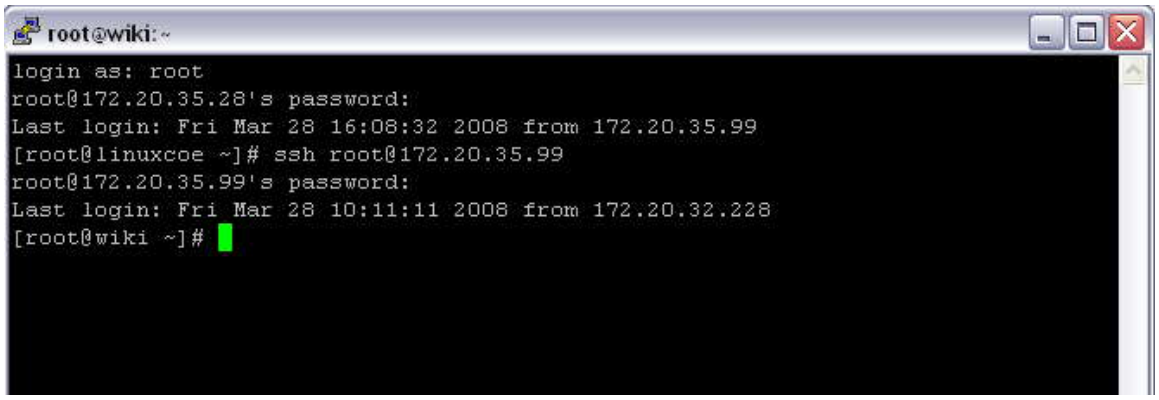
Output:



In this example iptables is being configured to allow the firewall to send ICMP echo-requests (pings) and in turn, accept the expected ICMP echo-replies.

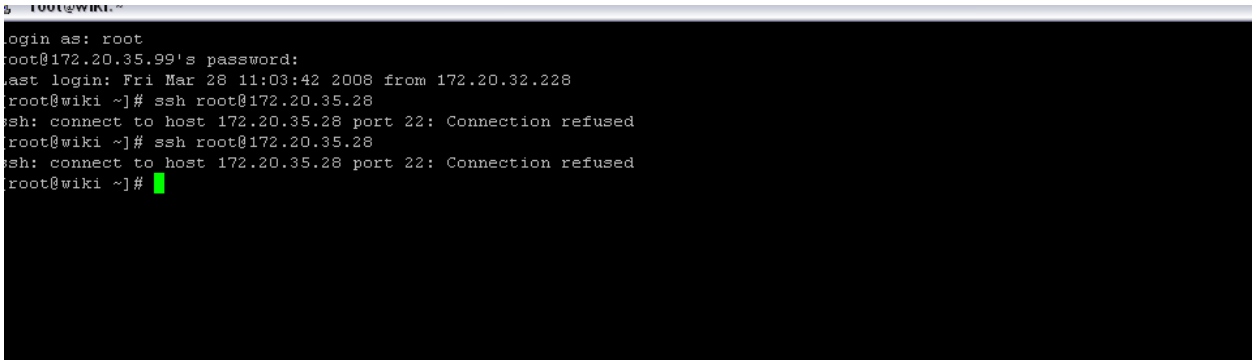
Scenario 3

In this scenario, we are able to SSH the wiki server from the Penrose system. I have applied the iptables rule to block the SSH in wiki server from the Penrose system



iptables -A INPUT -s 172.20.35.28 –d 172.20.35.99 –p TCP –dport 22 –j REJECT

Output



Saving Iptables

The service iptables save command permanently saves the iptables configuration in the /etc/sysconfig/iptables file. When the system reboots, the iptables-restore program reads the configuration and makes it the active configuration.

[root@linuxcoe]# cat /etc/sysconfig/iptables

.