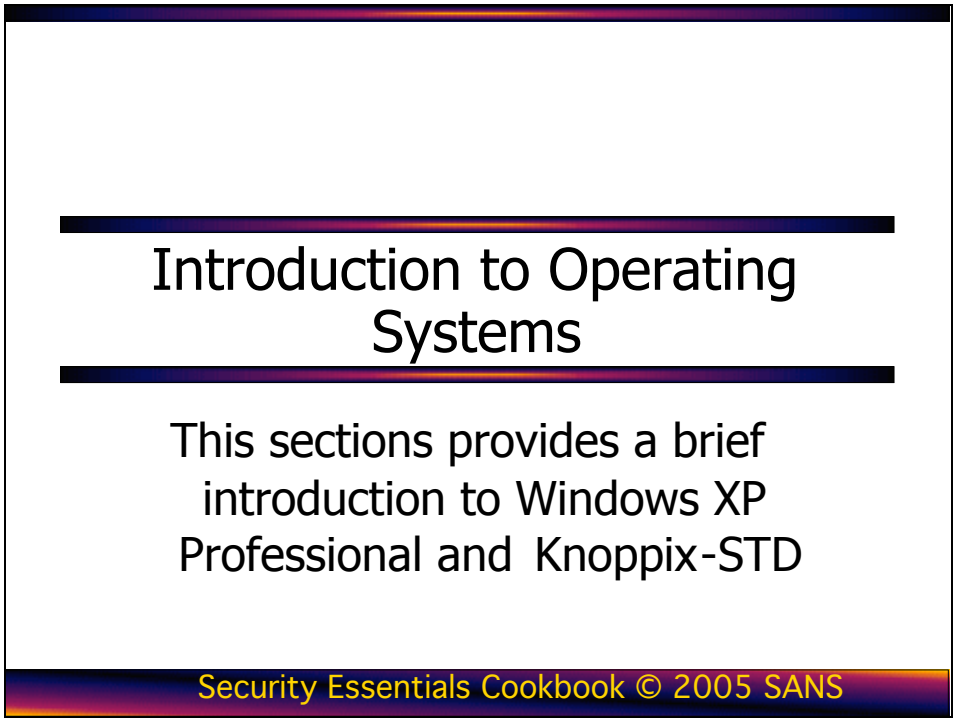
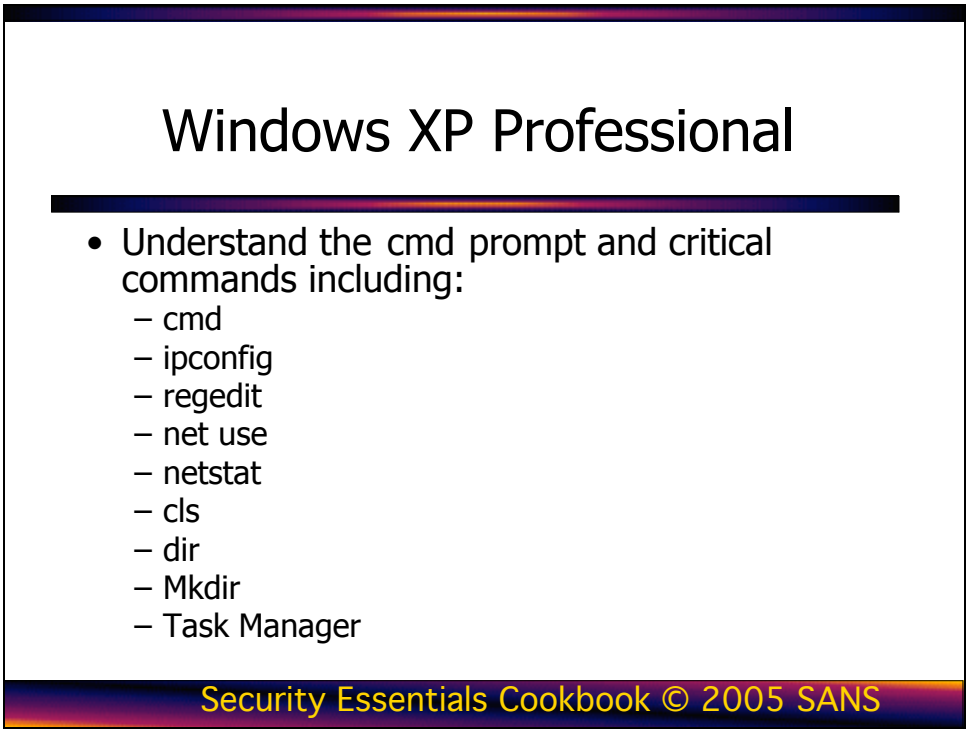


It is important that you familiarize yourself with Windows XP Professional and Knoppix-STD as preparation for this course. The exercises in this book assume a basic knowledge of both of these operating systems. This chapter provides an overview of both operating systems. It is not intended as a comprehensive guide to Windows XP Professional and Knoppix-STD; it is intended to help prepare you for this course.



Introduction to Windows XP Professional

The Windows XP operating system is a dynamic and continually changing operating system with new security patches and hot fixes being released often. In a normal production environment, it is highly recommended that you maintain a patching schedule to keep your systems up-to-date. For the purposes of this book, it is important that you do not patch your system. Because this system will be vulnerable to most of the exploits that have been discovered since Windows XP was first released, it is extremely important that you do not connect it to a production network. Several patches will cause issues when completing various labs in this book. By following the installation guide, you are assured of getting the maximum value out of the activities covered throughout this book.

This "Introduction to Windows XP Professional" guide teaches you about the basic commands and actions you need to know for the Security Essentials Boot Camp. This document introduces you to the following: **cmd, ipconfig, regedit, net use, netstat, cls, dir, mkdir, and the Task Manager.**

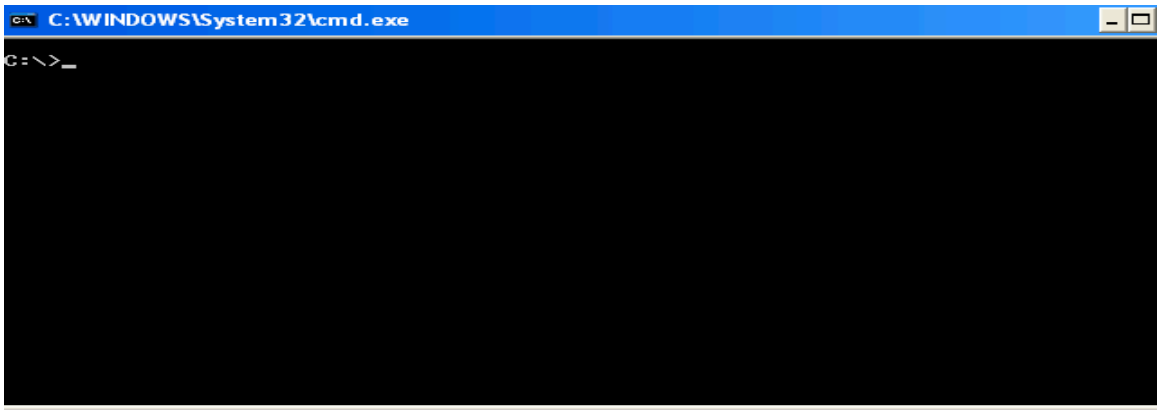
The 32-bit Cmd Prompt

Since the release of Windows 2000 Professional, the old 16-bit **command.com** program has been replaced with the 32-bit **cmd.exe program.** There are many benefits of using **cmd** including the following:

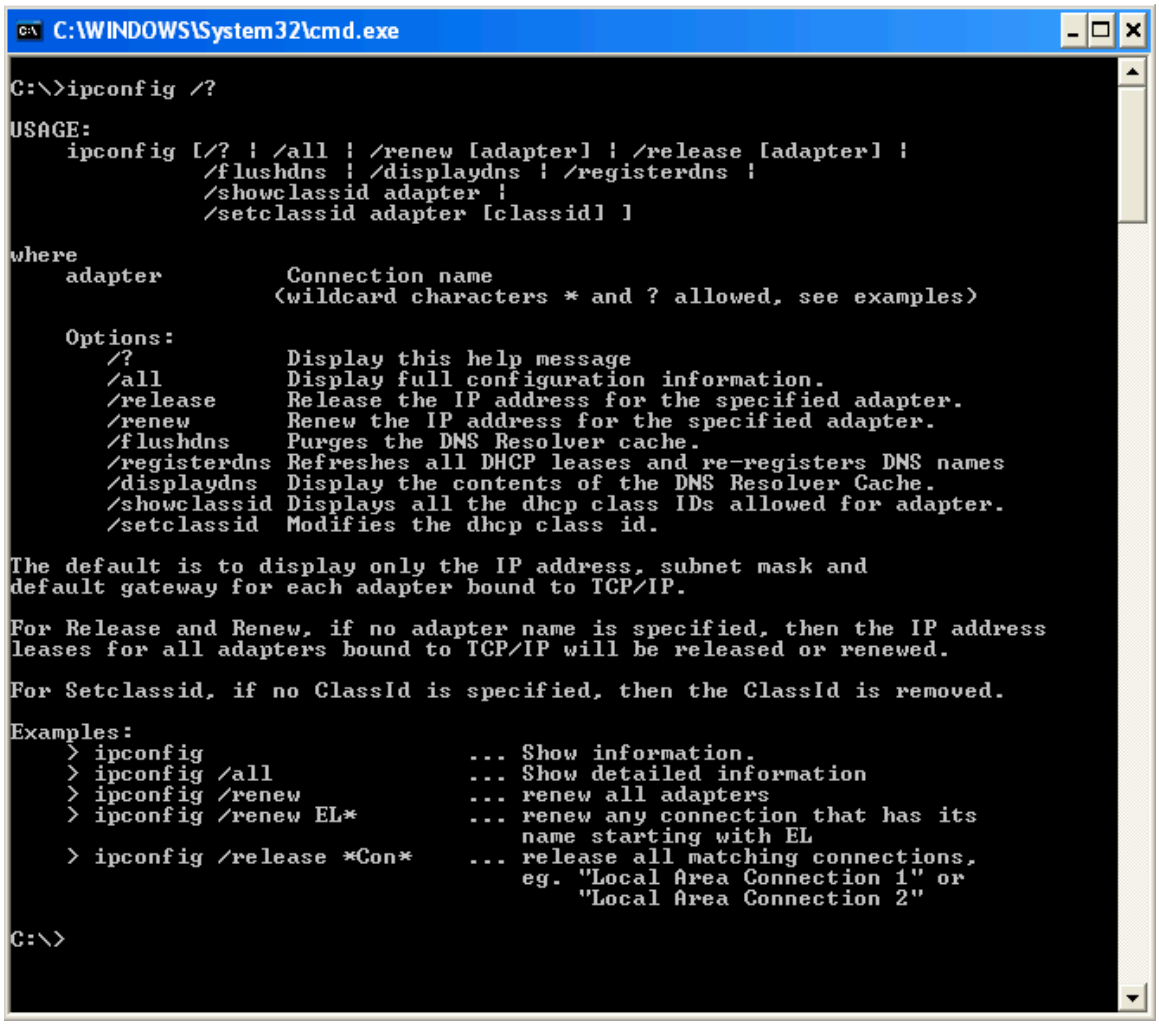
- The capability to run scripts in both the CMD language as well as the REXX language
- There are no 8.3 filename limitations
- The capability of running multiple commands on the same command line
- Support for command pipelines
- Help functionality with **/?**

The following list of tasks shows you how to use the command prompt to obtain help or information about your system:

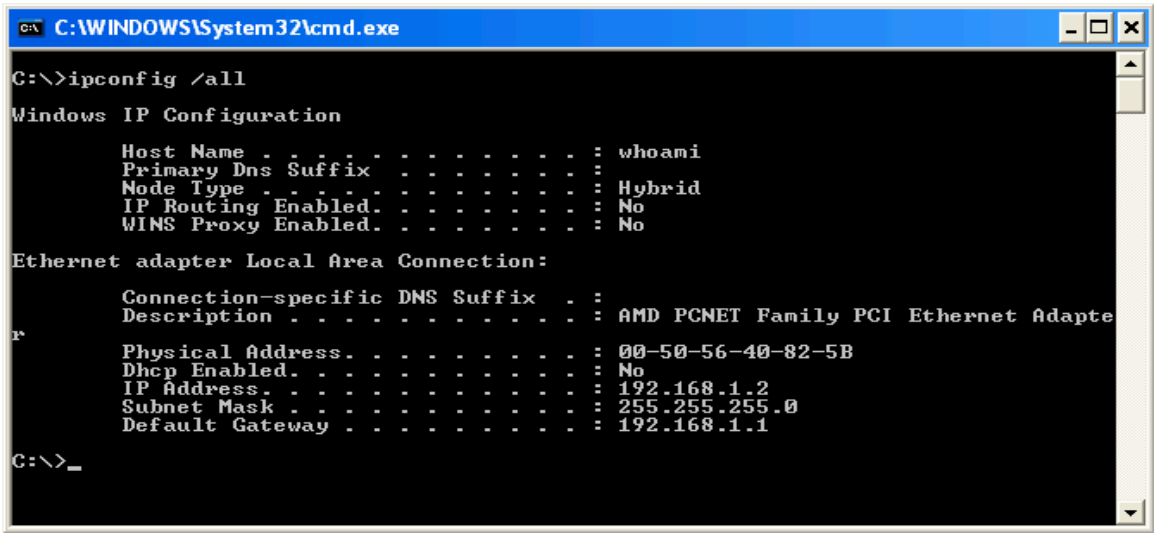
1. To display the command prompt, select **Start, Run**, and then type **cmd**. The following window appears.

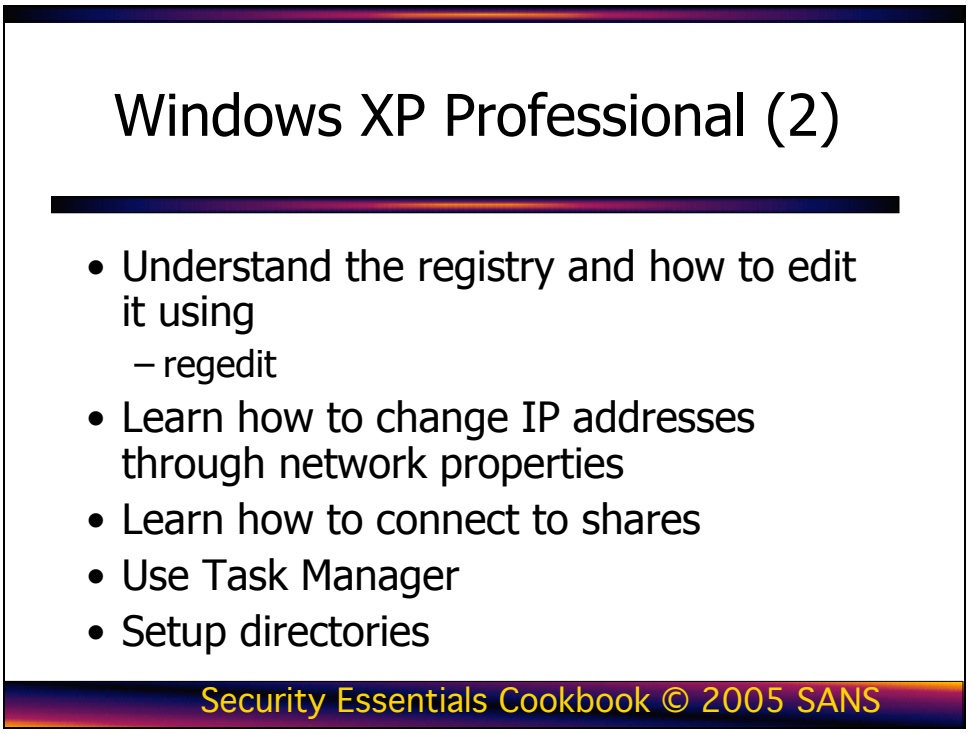


2. If you need help with a command while using **cmd**, type **/?** after the command in question. To get NIC TCP/IP information, type **ipconfig**. To get a list of the available ipconfig options, type **ipconfig /?** after the command prompt, as shown in the following screen.



3. To get the IP address information for your system, type **ipconfig /all.** This also displays your MAC address, as shown in the following screen.





This page intentionally left blank.

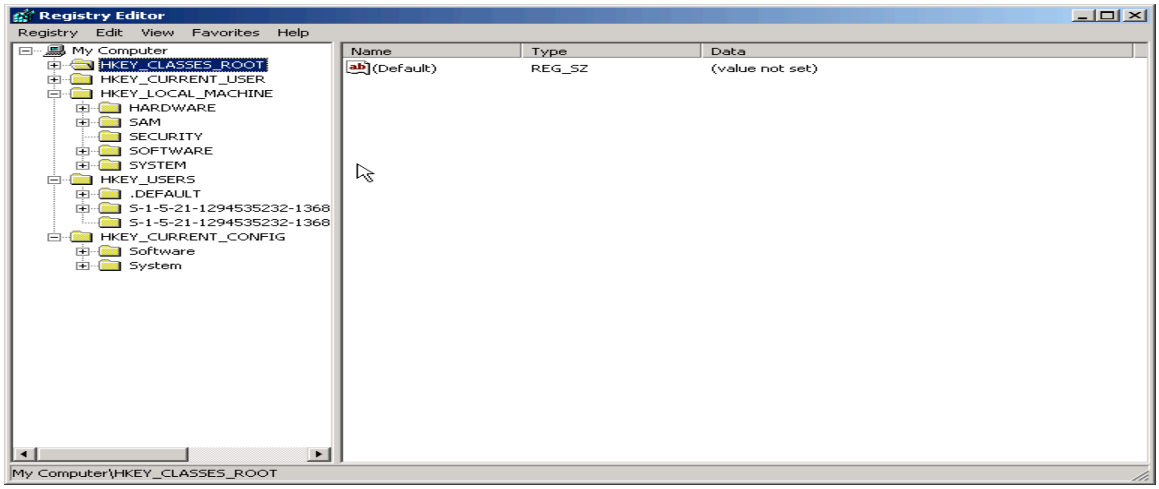
Editing the Registry

To edit the registry in a Windows environment you can use the **regedit** command at the Run prompt. One of the nice features of using **regedit** is that you can search every hive for specific keys, values, and data.

Warning: When using **regedit,** exercise extreme caution because any change you make is permanent and could potentially render your system unusable.

The following list of tasks explain how to edit the registry and how to use **regedit**:

1. To start **regedit**, choose **Start**, **Run**. Then, type **regedit** and press **Enter**. The following is a screen shot of **regedit**.

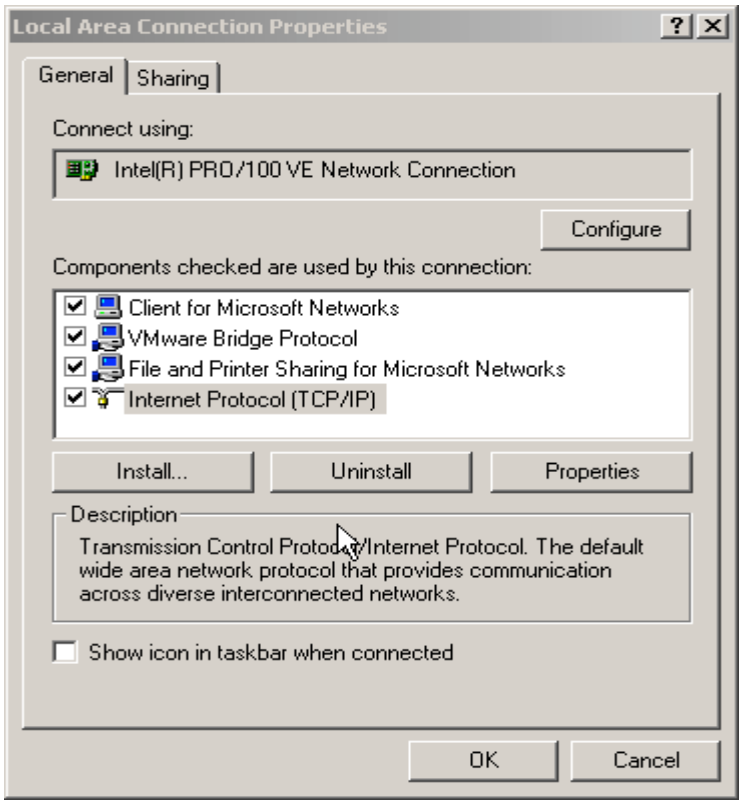


2. To ensure you can recover from making detrimental mistakes when editing the registry, you should always save a copy of the keys you change. To do this, choose **Registry, Save Key** and make a backup copy of the key.

IP Changes

The following steps are necessary for making IP changes on your Windows XP Professional system:

1. To make IP address changes to your local machine, open the NIC properties. Open up your Control Panel by choosing **Start, Control Panel, Network and Internet Connections, Network Connections**. Highlight the local area connection. Right-click the Local Area Connection, and click **Properties.** The following screen appears.



2. Highlight **Internet Protocol (TCP/IP)** and click the **Properties** button.

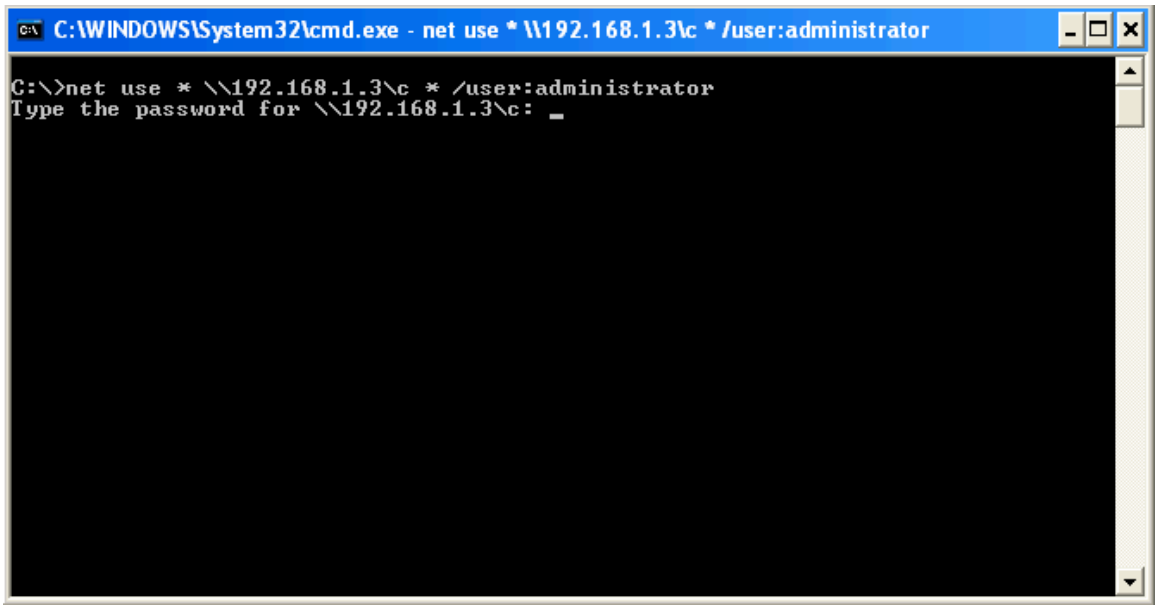
Connecting to Remote Devices

If you are like me, you don't want to browse to a remote device to connect to it. To make connections to remote devices without browsing to them, follow these steps:

1. Make a direct connection to a remote share by using the **net use** command. A typical **net use** command looks like this:

   Net use * \\<IP_ADDRESS>\<Share_Name> * /user:<Remote_User_Name

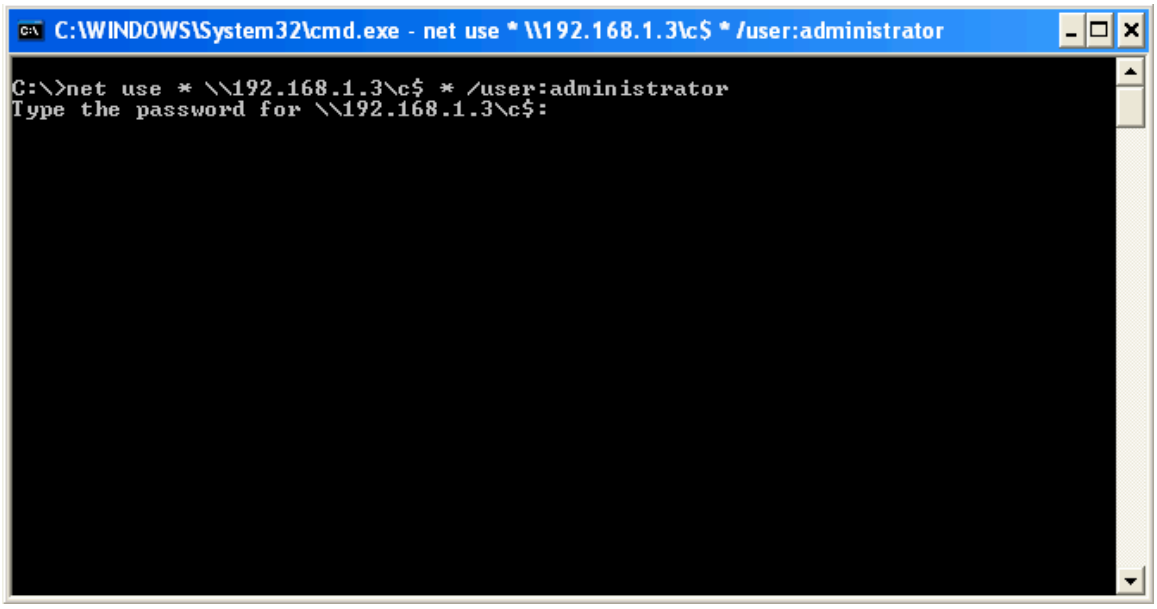
   Type the command after the prompt. Press **Enter**. You are prompted for a password, as shown in the following screen.



2. Even if a user does not explicitly share a folder or drive on his or her system, there are hidden administrative shares you can connect to if you happen to have the remote machine's Administrator password. Default administrative shares are as follows:

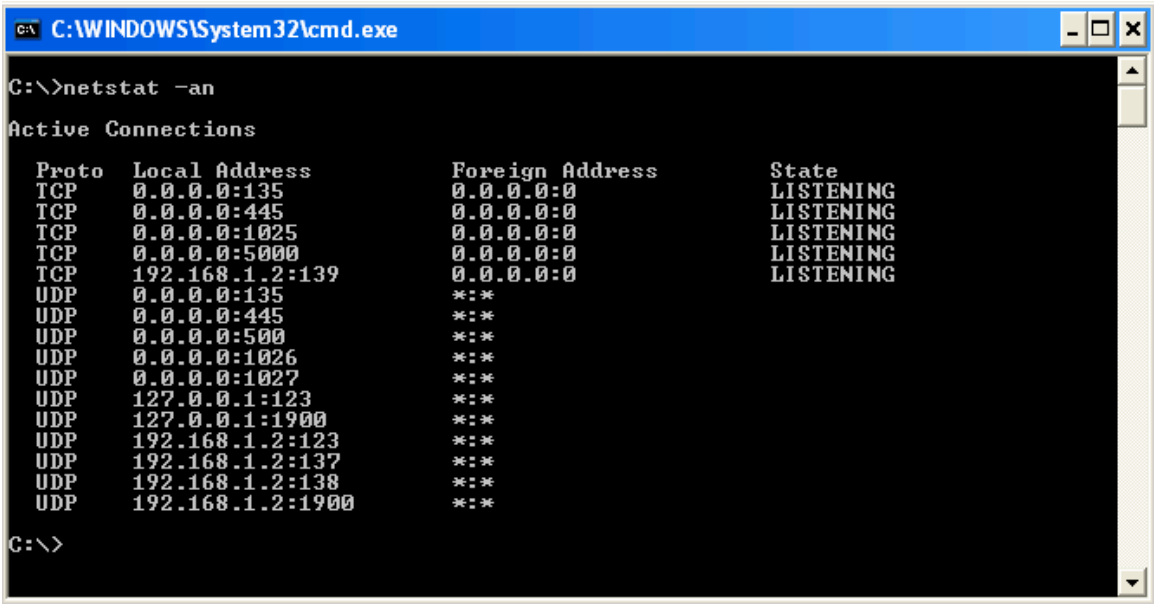
   C$, D$

3. To connect to the **C$** or **D$** share, simply replace the share with the desired administrative share. The following screen shows how this was done for the share shown in the previous screen.



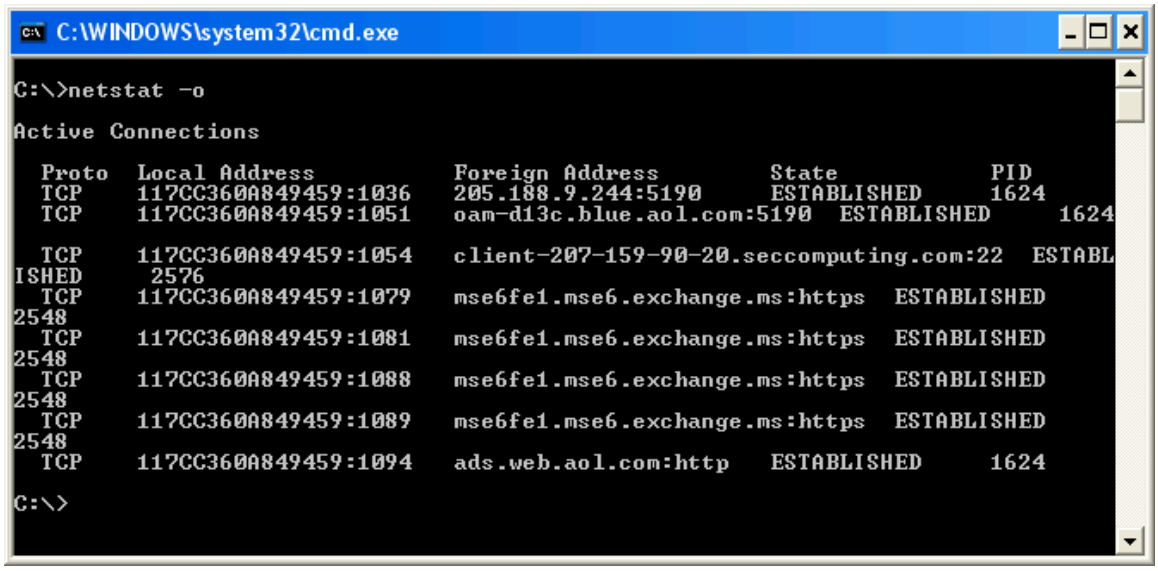
Viewing Ports

Throughout the exercises in this book, you are required to add services to your system. To see what ports are open on your box, you can use the **netstat** command, as shown in the following screen.



You can see every open port and the state of each port. States include listening, waiting, or connected. The **netstat** command also shows you TCP and UDP connections.

While netstat will show you which ports are open, by default it does not show you which service is causing a given port to be open. Attackers connect to systems via ports. The more ports that are open, the more avenues of attack. Therefore it is important to shutdown unneeded ports. In order to close a port you need to know which service is causing a given port to be open. By typing the following command: **netstat –o** will show which service is causing a given port to be open.



Other Useful Commands

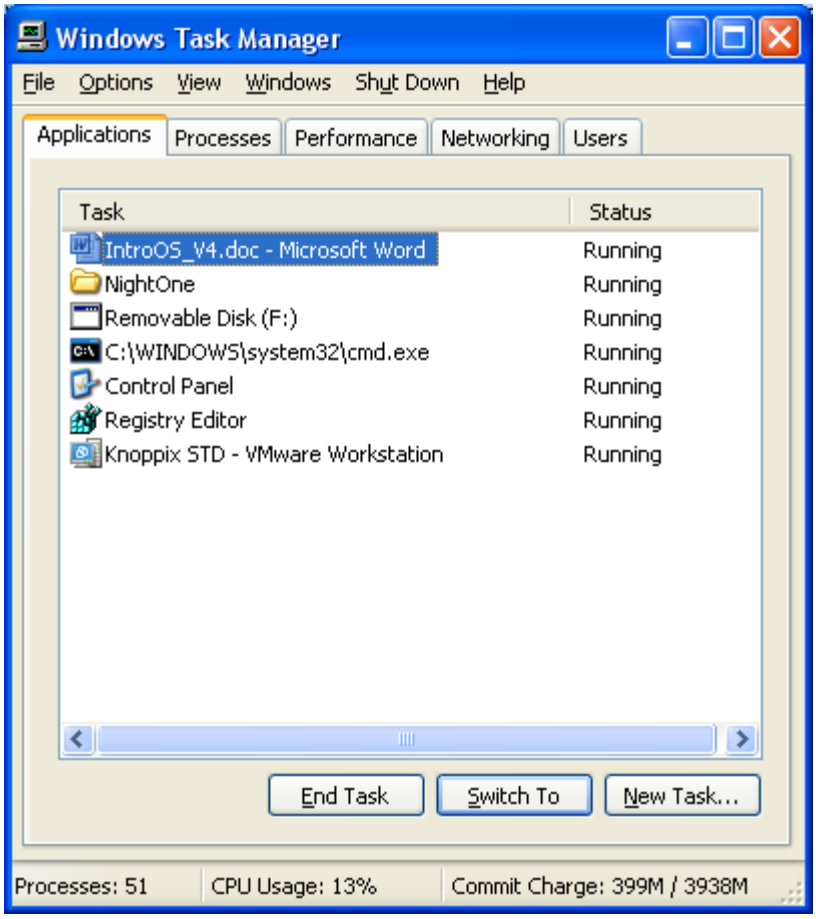
Some other commands to use at the cmd prompt are:

- **cls**—Clears everything on the screen and returns you to the top of the **cmd** window
- **dir**—Displays a directory listing
- **cd **—Returns you to c:\ from whatever directory you are in

Task Manager

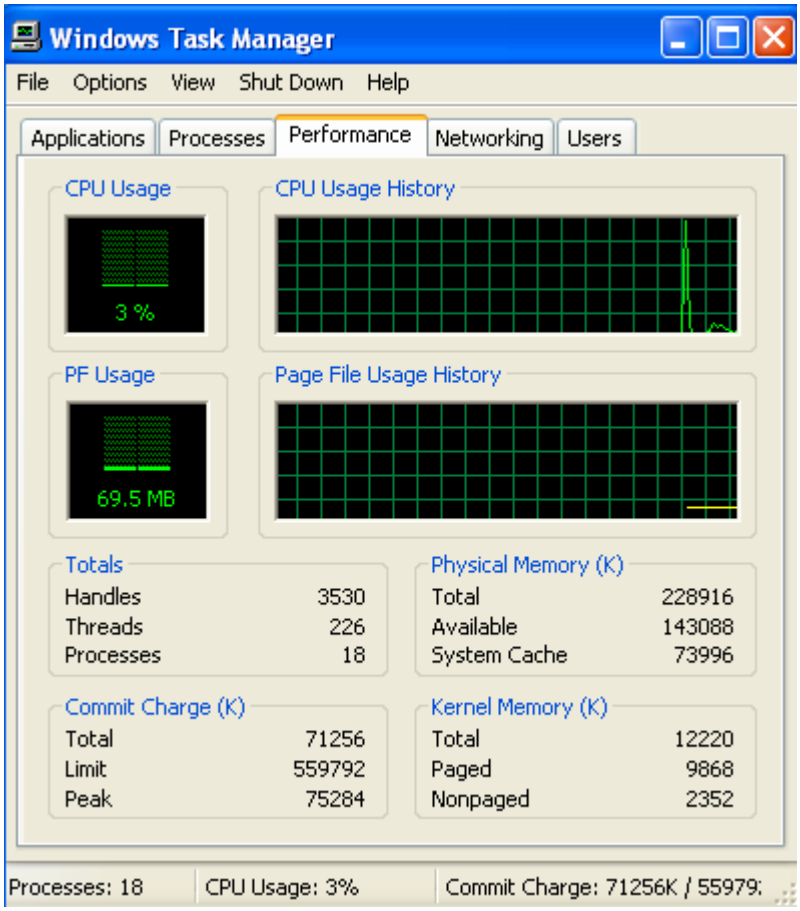
Another great built-in tool in Windows XP Professional is the Task Manager. The following list shows you how to open and use the Task Manager:

1. To open the Task Manager, right-click on the task bar and choose **Task Manager**. The default screen that opens shows which applications are running on the system.



Tip: Another way to open the Task Manager, which I recommend, is to hold down the **Ctrl, Shift,** and **Esc** keys simultaneously.

2. From this window, you can check the running processes on the device, the performance trends, and the applications that are currently running. It is a great tool to open if you have an application that stops responding. You can open Task Manager, highlight the application, and then choose to close the offending application. By clicking on the performance tab, you can see CPU and memory usage.

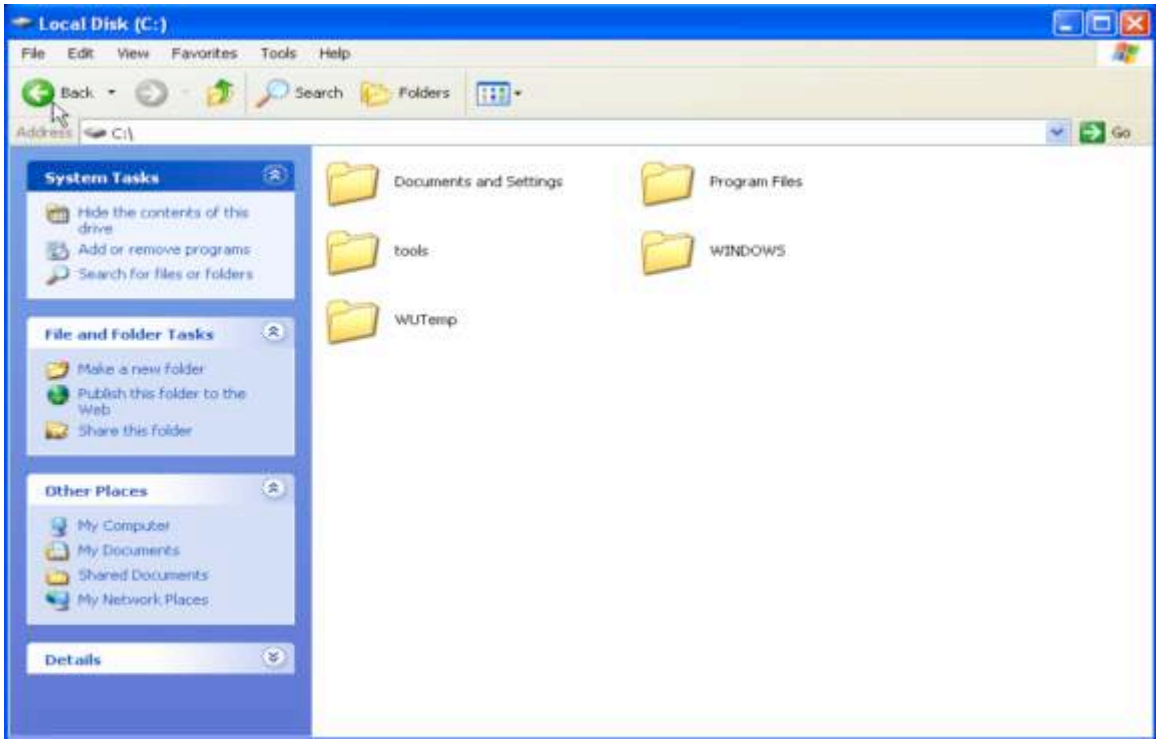


There are many other Windows XP Professional functions that are not covered in this book. Our goal is to give you the basics, so that you can quickly install and run the tools covered throughout the following chapters.

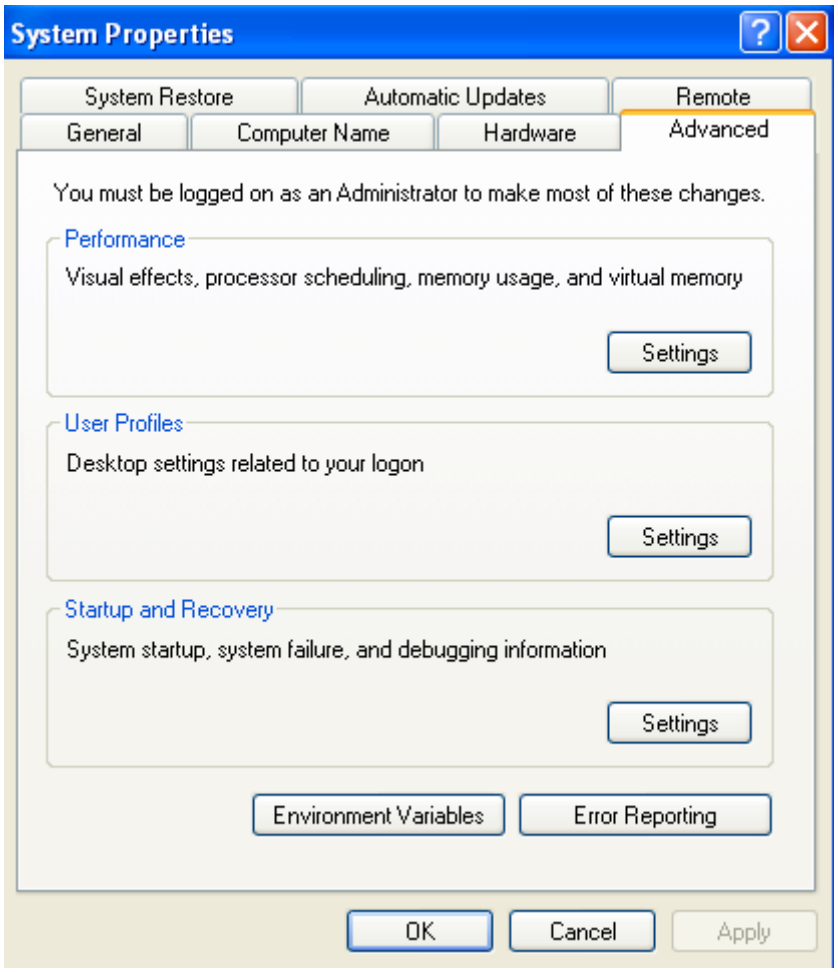
Setting Up the Directory Structure

Now that you have installed the operating system and seen some of the tools that are built into it, you need to set up your directory structure, so that it's consistent with the directory structure used for installing and storing the tools discussed in this book. Follow these steps to setup the directory structure:

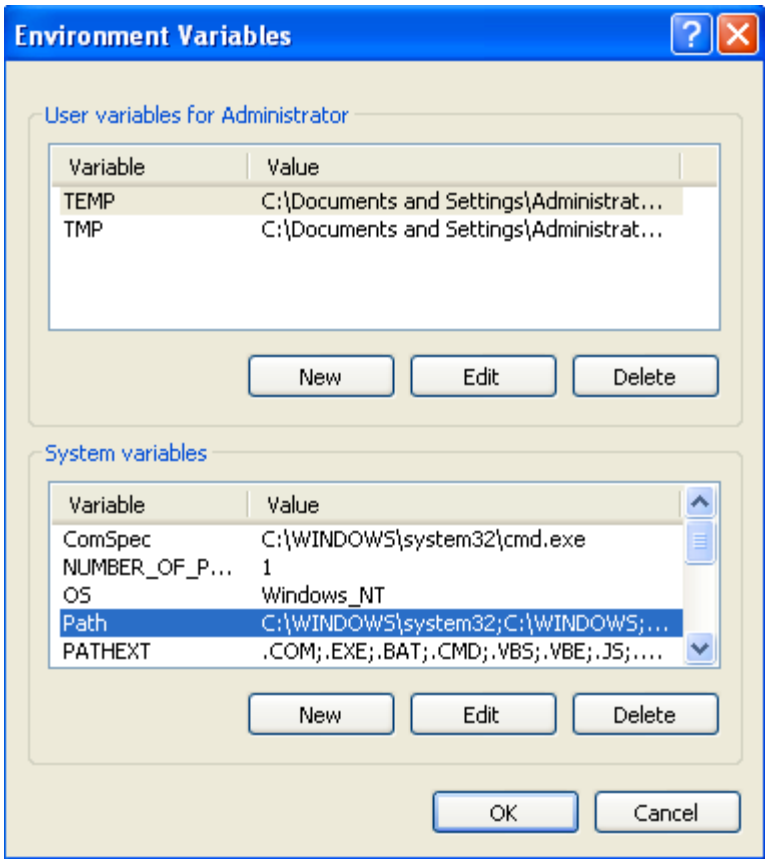
1. First click on **Start**, **My Computer** and then double-click **Local Disk (c:)**. The window that appears lists the structure of the **C:** drive.
   Right-click a spot in the window that is blank. Move your mouse down the menu that appears and left-click **New** and choose **Folder.** Name the folder **tools**. The following screen shows a directory structure with the new **tools** folder.



2. The exercises in this book require you to run several tools from the command line. Thus, you need to add the new folder we created, **c:\tools**, to PATH. If you do this, you won't have to navigate to the **tools** folder each time you want to run an application. To add the folder to PATH, Click on **Start**, right-click on **My Computer**, and choose **Properties** from the pop-up menu. Click the **Advanced** tab, as shown in the following screen.



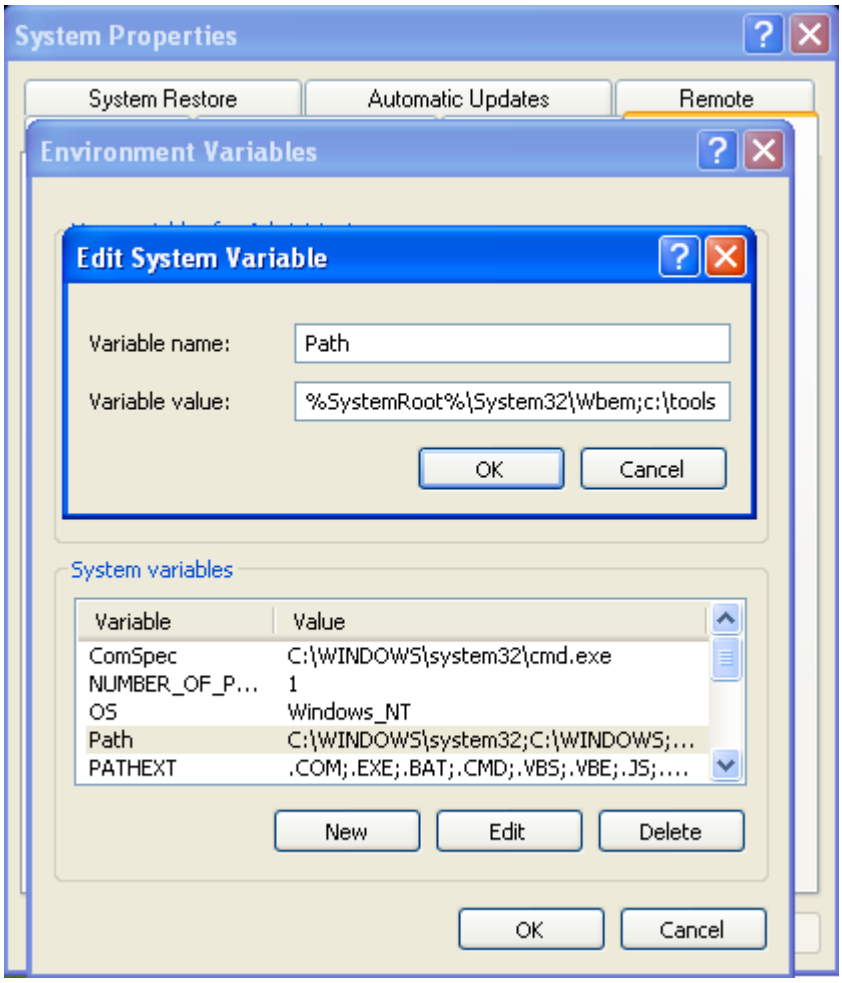
3. Click the **Environment Variables** button**.** In the **System Variables section,** highlight the line labeled **Path** and click **Edit**.



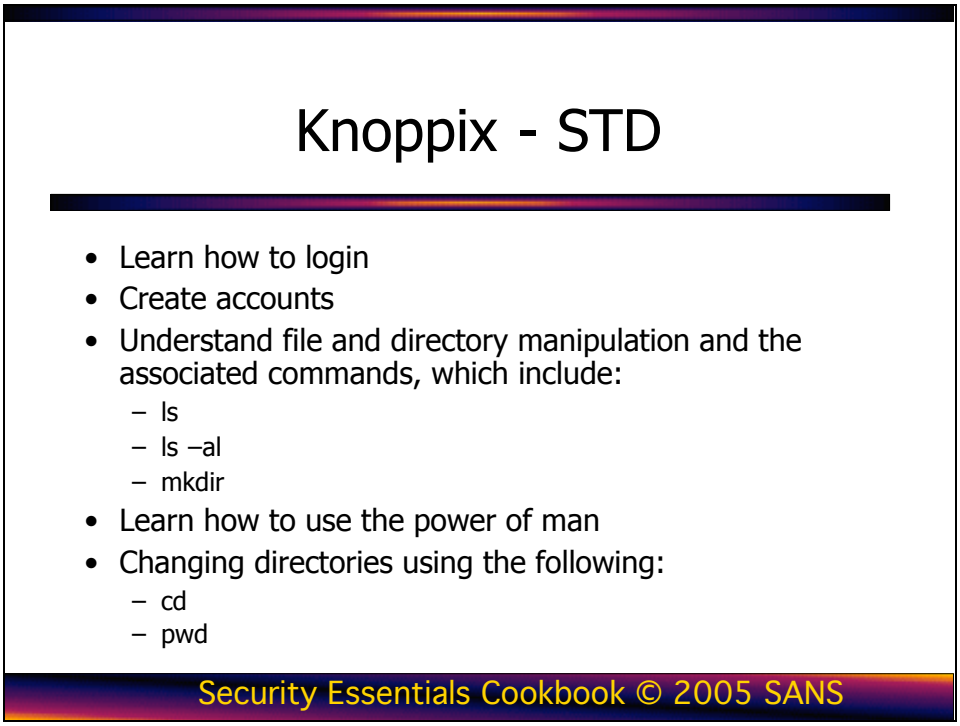
4. In the **Variable Value** field, move your cursor to the end of the line and add the following exactly as it is shown here:

;c:\tools

Click **OK** on each of the **Edit System Variable**, **Environment Variables** and **System Properties** windows.



The executables located in **c:\tools** can now be run from any directory in your file structure. This saves a lot of time when you are using command prompt and want to run an application from it.



Introduction to Knoppix-STD

Linux is an open source operating system that runs on a wide range of hardware platforms. So what is open source, you ask? As an open source system, Linux is protected under the GNU General Public License, which guarantees the freedom to use and change the software it covers. Numerous Linux distributions are available from many companies, and each distribution has its own advantages and disadvantages. With these characteristics comes a faithful user following who think that *their* preferred distribution is the best. Some of the Linux distributions that are currently available include Red Hat, S.U.S.E., Debian, and Mandrake.

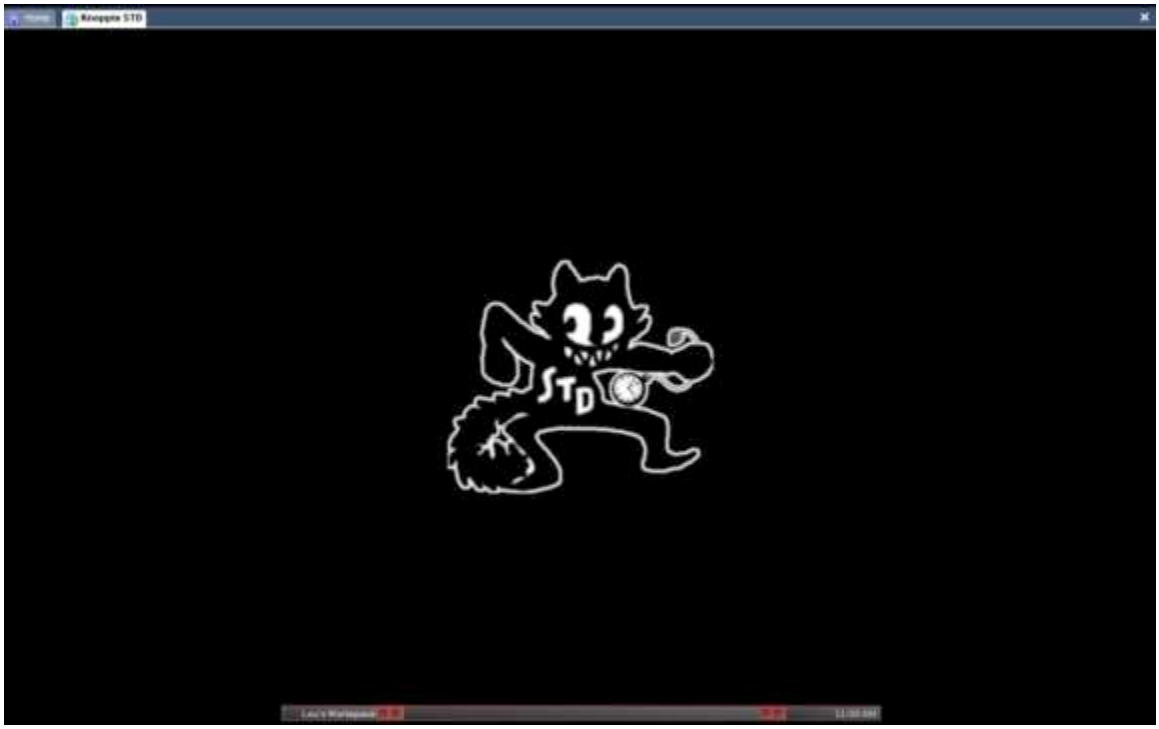
As Linux became popular, a bootable version of Linux emerged called Knoppix. Knoppix resides on a CD and can be booted without disturbing the current OS that is on your hard drive. Based on the flexibility, Knoppix is the recommended Linux distro for this class. In addition, since we are concerned with security, Knoppix-STD is the version we are going to use.

At the heart of each distribution is the kernel, which behaves like a crossing guard. The kernel handles such functions as memory management, security, and resource allocation. The kernel also provides features such as true multitasking, threading, and TCP/IP networking. Contrary to popular belief, the kernel is, in fact, Linux. All other applications and programs are part of a particular distribution.

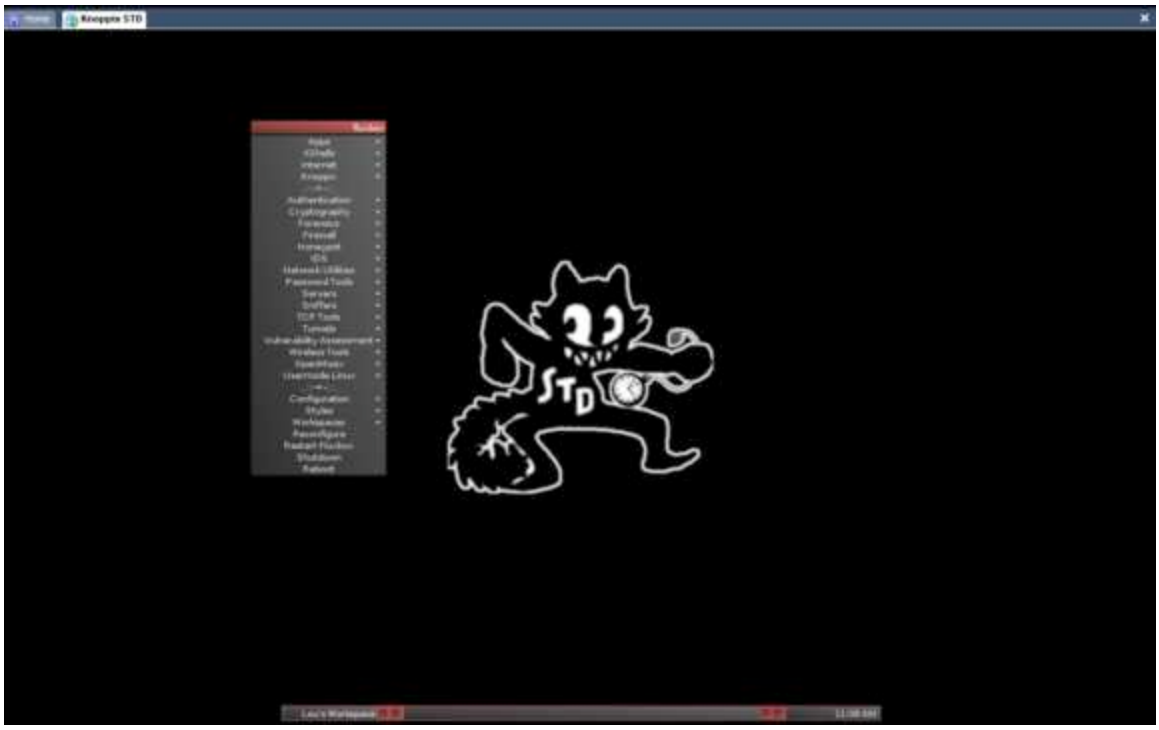
The Linux *shell* is another name for the command shell, which is similar in function to a DOS shell. It is the program that gives you an interface to type commands, and it accepts the commands you type. During the examples that follow, remember that nearly everything in Linux is case sensitive.

Starting-up Knoppix

This section describes how to start-up Knoppix on your system. Insert the Knoppix CD into your computer and reboot. Make sure you select the option on your computer to boot off of the CD. After booting off the Knoppix-STD CD, the main screen appears. There is no need to log on.

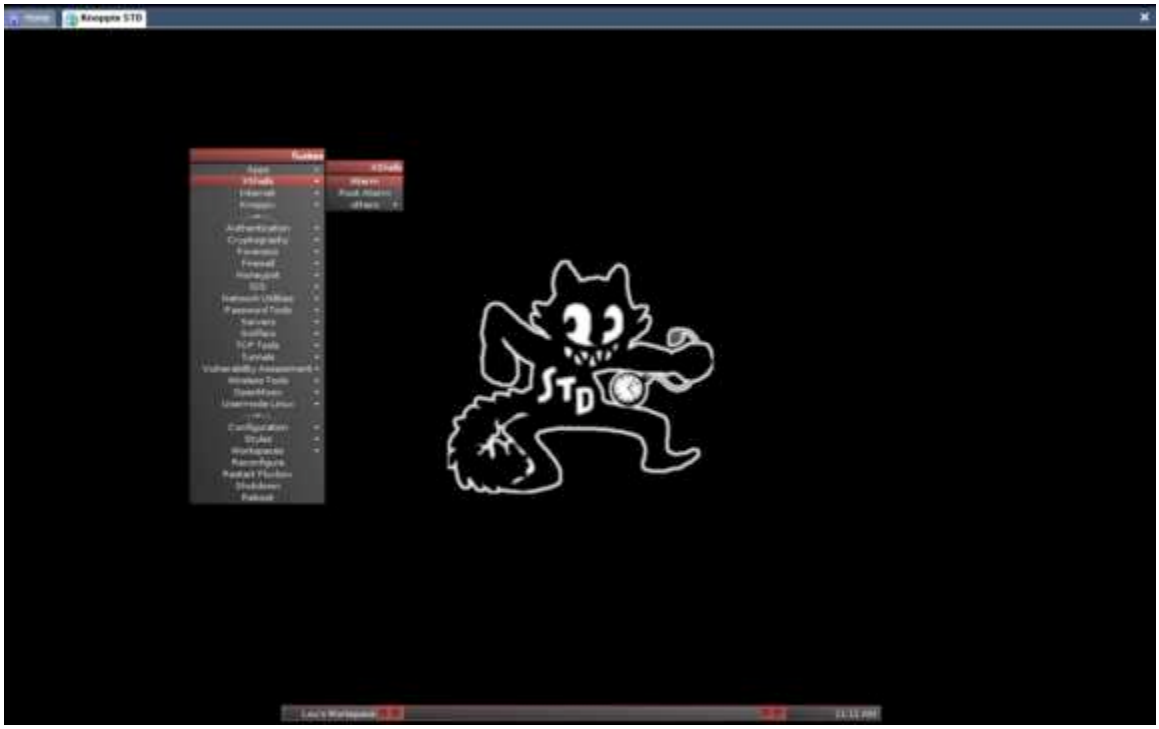


To get a list of all of the programs that are installed, right click on the main screen and a pop-up menu will appear.



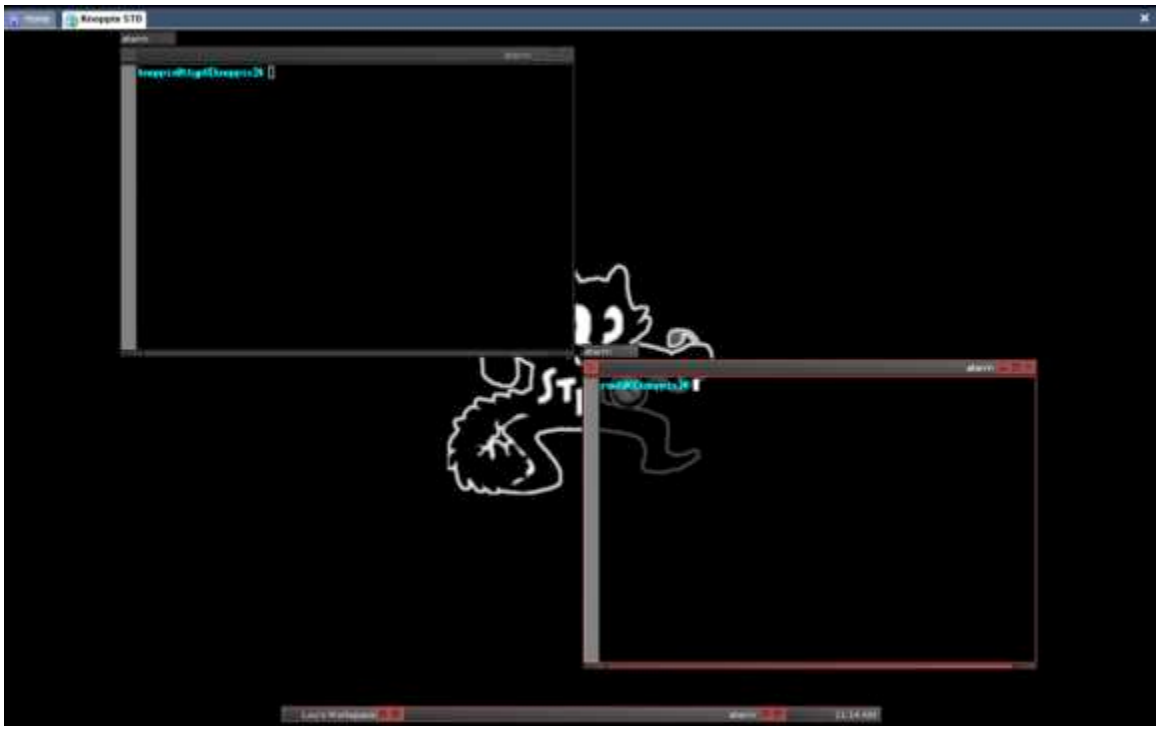
You can see that all of the security tools you would use are already installed.

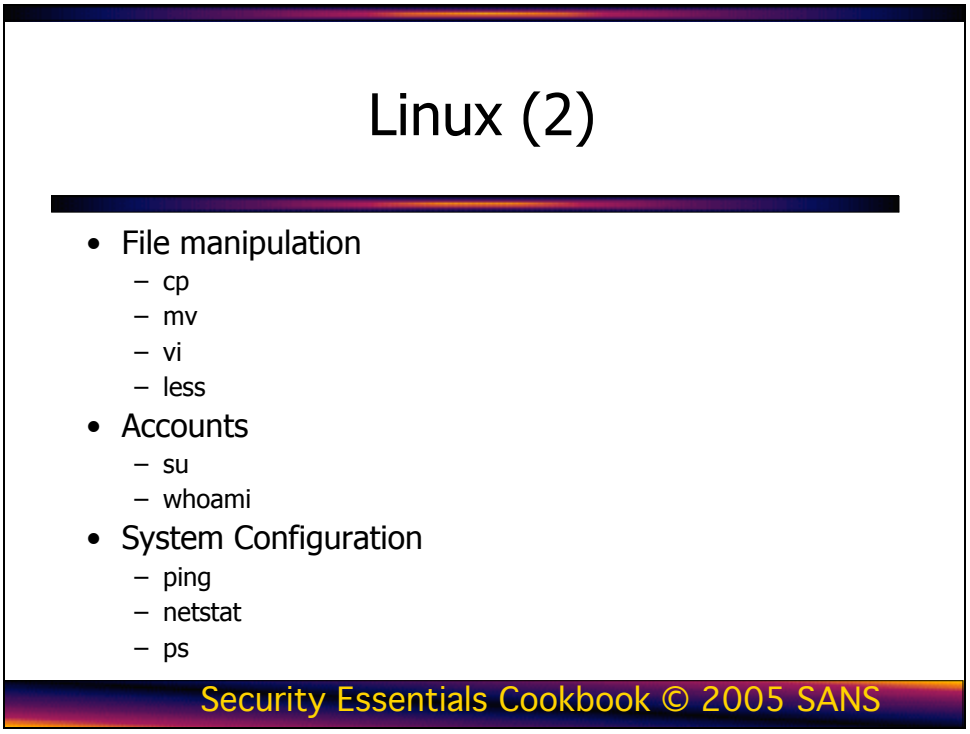
To type commands on the system you will need to open a terminal window. To open a terminal window, right click on the desktop, select XShells and then select Aterm if you want a standard shell or Root Aterm if you want a root shell.



A standard shell allows you to type normal user commands. A root shell allows you to type privileged commands. To adhere to a principle of least privilege it is usually a good idea to log in as a normal user and then type **su** to upgrade to root privileges.

When you are logged in as a normal user, your prompt is listed as $. When you are logged in as root your prompt is listed as #. In the following screen shot, the upper left is a normal user and the lower right is root.



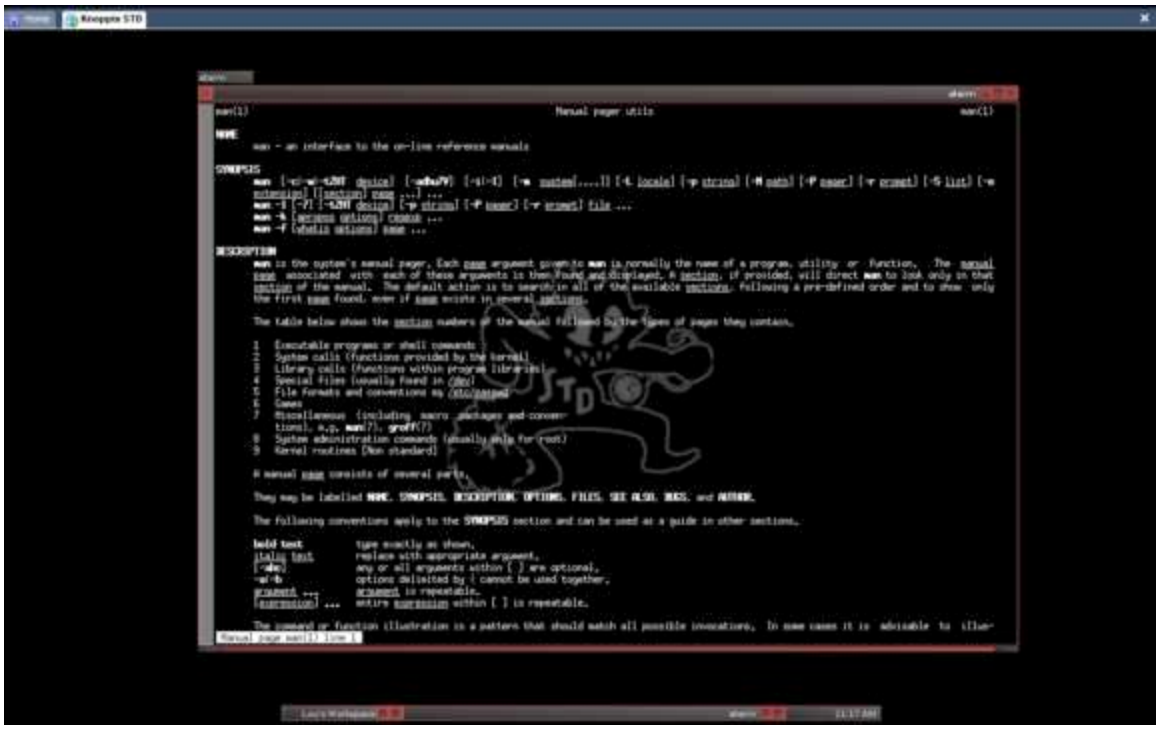


File and Directory Manipulation

This section introduces you to the basics of Linux by covering some of the most common commands, files, and directories used in Linux. Each topic includes a brief description and an example of how the topic is used. You can find more information on each topic by typing **man <topic>**. For example, issue the following command at a shell prompt:

man man

This command displays a manual that describes the **man** command and also demonstrates how man pages are formatted.

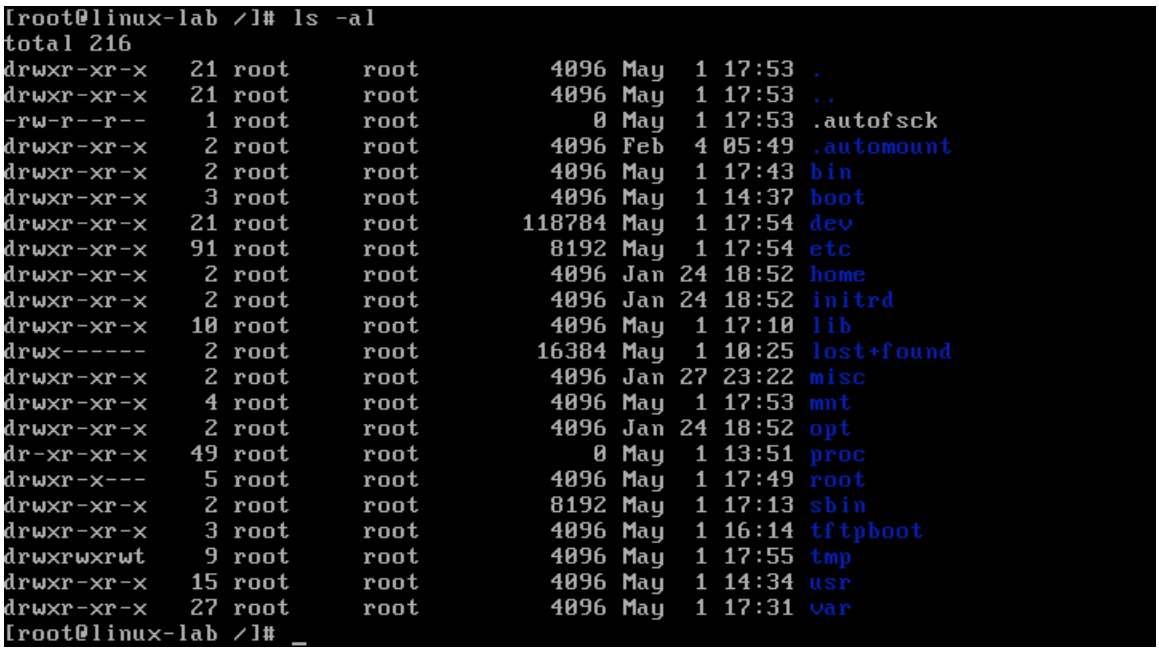


The ls Command

One of the most basic commands in Linux is **ls**. Just as you use **dir** in DOS, type **ls** to output a listing of the directories and files that are contained within the current directory. Open a root shell as described above. Now, in the command shell, type **ls**, as shown in the following screen.

```
[root@linux-lab /]# ls
bin   dev  home    lib         misc  opt   root  tftpboot  usr
boot  etc  initrd  lost+found  mnt   proc  sbin  tmp       var
[root@linux-lab /]# _
```

As with most commands in Linux, you can specify options to change the result of the command's execution. For example, enter **ls –al** in the command shell, as shown in the following screen.

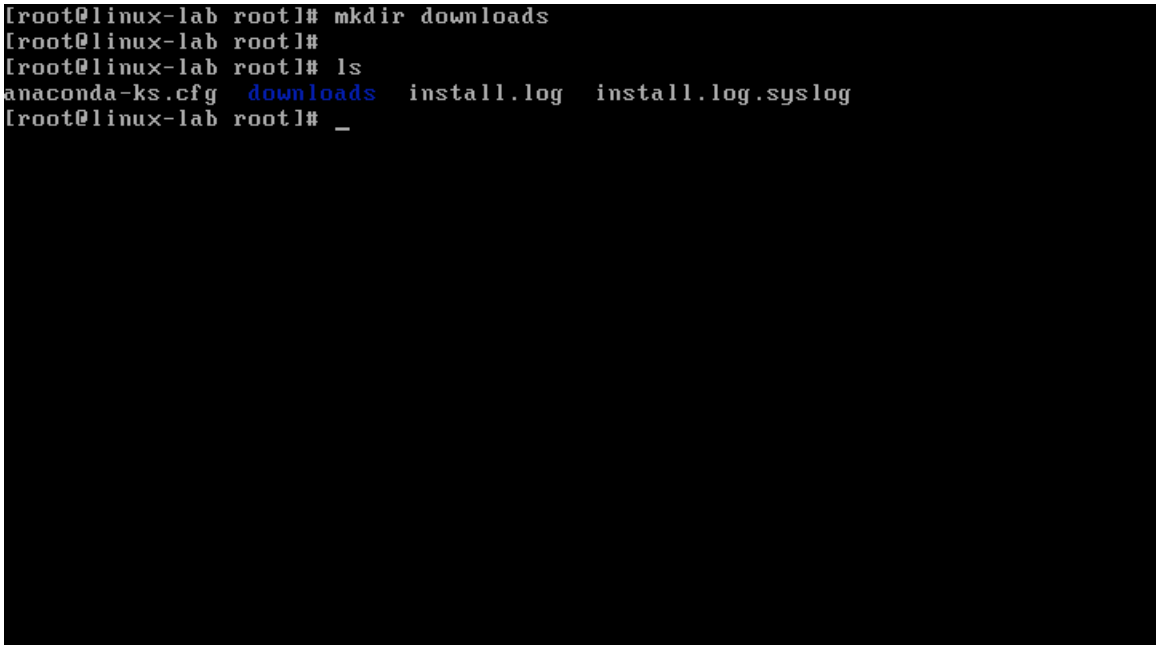


The **–a** option tells the command interpreter to show all files, and **–l** tells it to use the long listing format. These are two of the many options that can be used with **ls**.

Changing Directories and Creating Directories

Now that you can tell what files and directories the root directory contains, let's move back to the /root directory. In order to change directories, you use the **cd** command; here, you would use the command **cd /root**, as shown in the following screen.

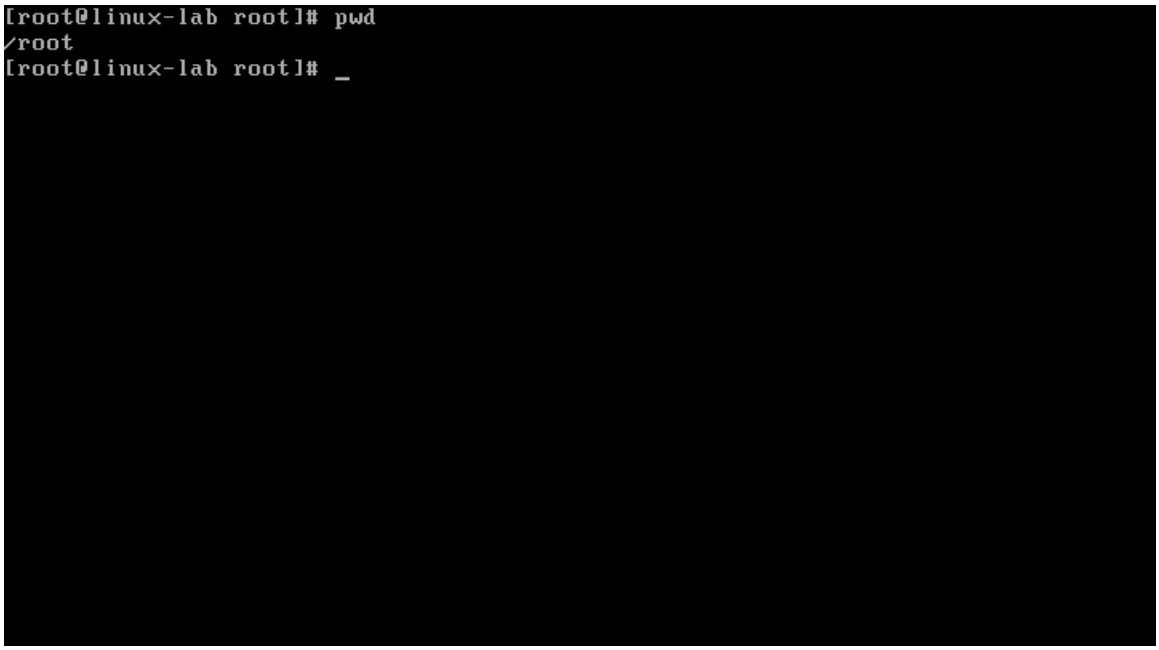
You can use the **mkdir** command to create a directory. The format of the **mkdir** command is **mkdir <directory_name>**. For example, type **mkdir downloads** to create a location where we save files that have been downloaded from the Internet.



Issuing **ls** after the **mkdir** shows the newly created download directory, as shown in blue in the previous screen. Note: Your directory will not list any files.

Determining Directory Placement

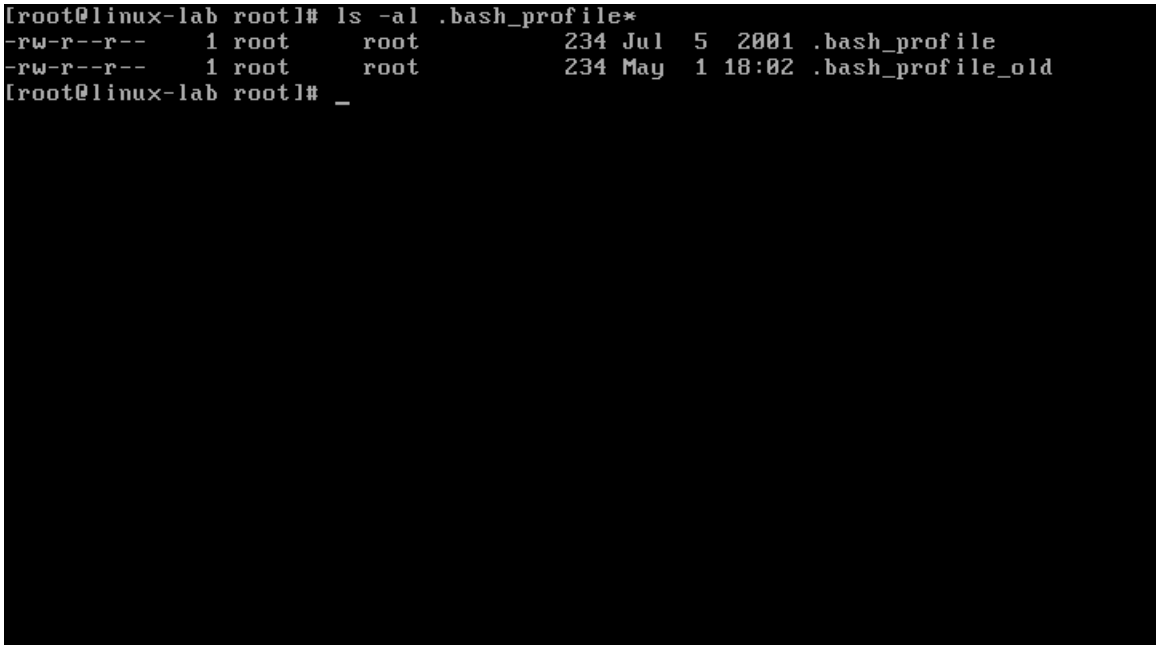
After using **cd** and **ls** to learn about the Linux structure of changing directories, you may not remember which directory you are currently in. You can determine where you are in the directory structure by typing **pwd** (print working directory).



Copying, Naming, Moving, and Removing Files

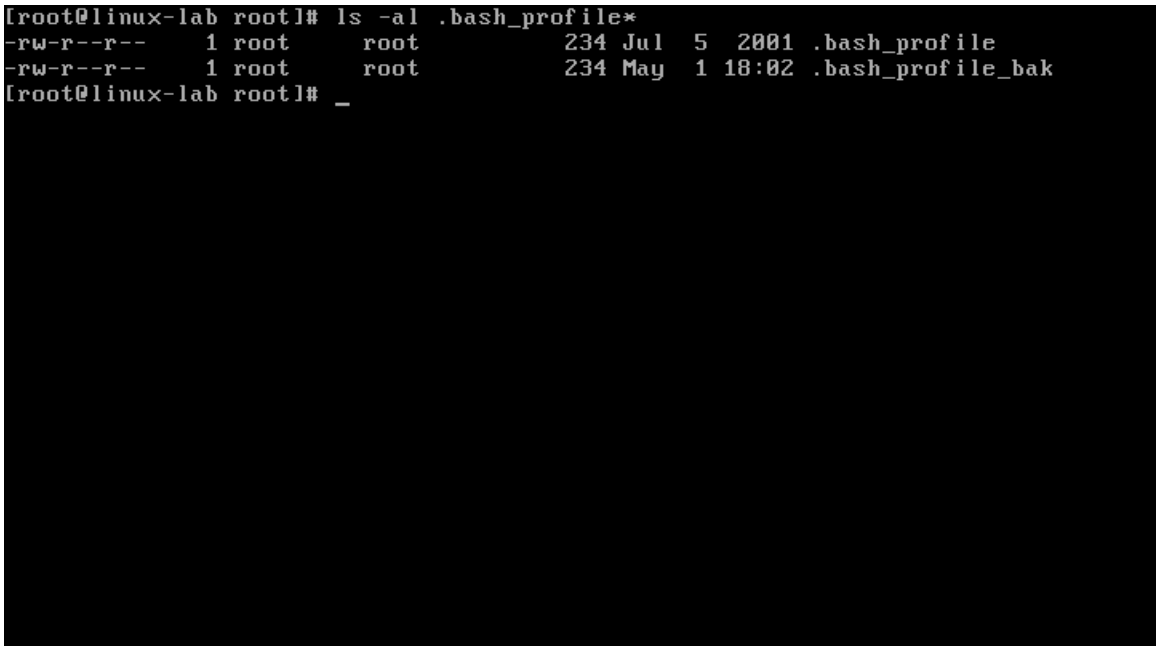
Now that you've logged in and moved around the file system, you can use the **cp** command to copy a file from one location to another. First, make a backup copy of the **.bash_profile** file named **.bash_profile_old**.

If you were to issue the command **ls –al .bash_profile***, you would see that there are now two copies of the **.bash_profile** file in the /root directory.



If you want to change the file's name or the file's location but do not want the original file to remain, you can use the **mv** command to move one file to another location. This is shown in the following screen.

Issuing the **ls –al .bash_profile*** command again will reveal that there are still two copies of **.bash_profile** in the directory; however, you'll notice that **.bash_profile_old** is now **.bash_profile_bak**.



When you no longer need a file, you can use the **rm** command to remove it. In this case, you could remove the **.bash_profile_bak** file by issuing the command **rm .bash_profile_bak**. When asked if you want to remove the file **.bash_profile_bak,** type **y** and press **Enter**.

Altering and Creating Files with the Vi Editor

You can use the **vi** editor to alter or create a file. Follow these steps:

1. Issue the command **vi <filename>** to create a file. For example, type **vi linux_lab** into a command shell. To enter insert mode press **i**.

2. Now type the string **Linux file creation using vi**, as shown in the following screen.

3. Press **Esc**, then press **Shift** + **:** to save the file**.** Then, type **wq** ('w' writes the file and 'q' quits vi and returns to the command shell), and press **Enter.**

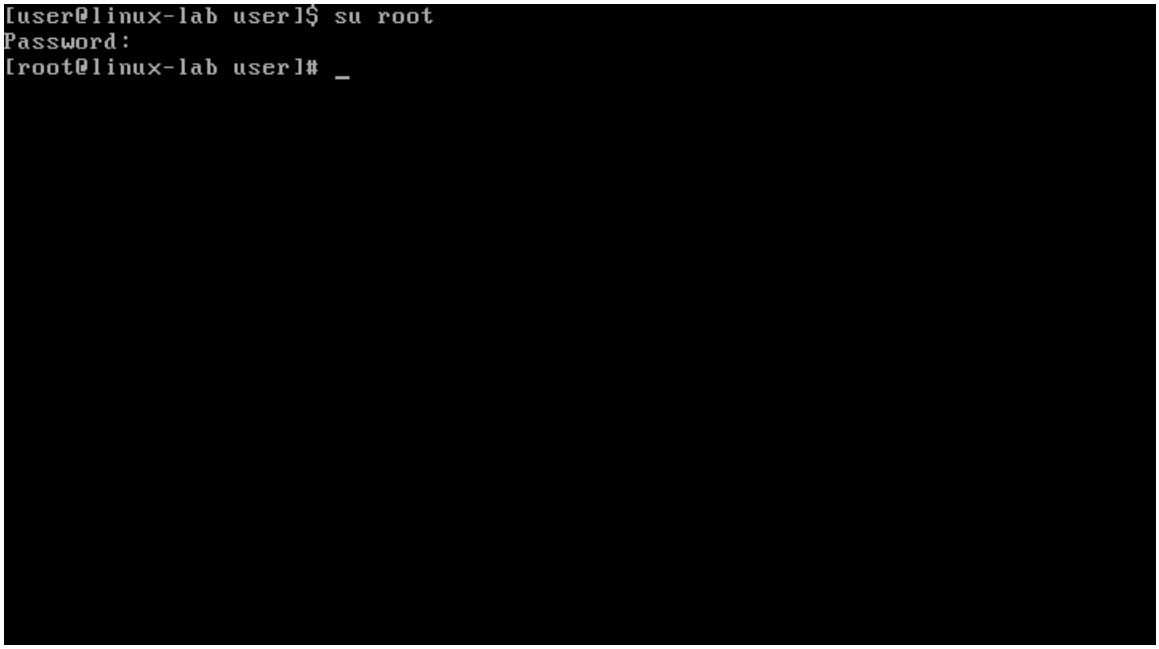
Viewing Files

There are many ways to view the contents of a file in Linux. One command you can use for this function is **less**. To view the contents of the **linux_lab** file that we just created, follow these steps:

1. Issue the command **less linux_lab**, as shown in the following screen.

2. The arrow keys allow you to navigate through the contents of the file. When you have finished, type **q** to exit and return back to the command shell.

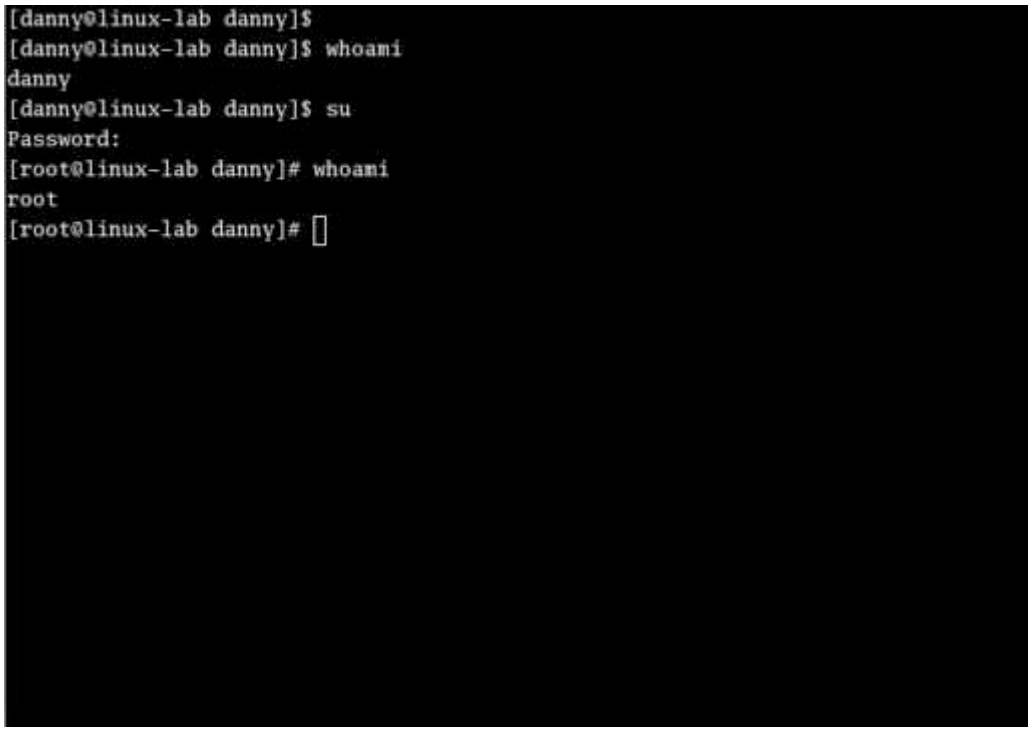
The ability to run some commands and view certain files is restricted to **root** for security reasons. When logged in to a Linux system as a normal user, you can escalate your privileges to **root** by using the command **su**. When logged on as a normal user, type **su** .



Notice that the last character in the shell prompt changed from **$** to **#**. This indicates that you are now logged in as **root**. To return to the normal user account, type **exit**.

Determining Account Types

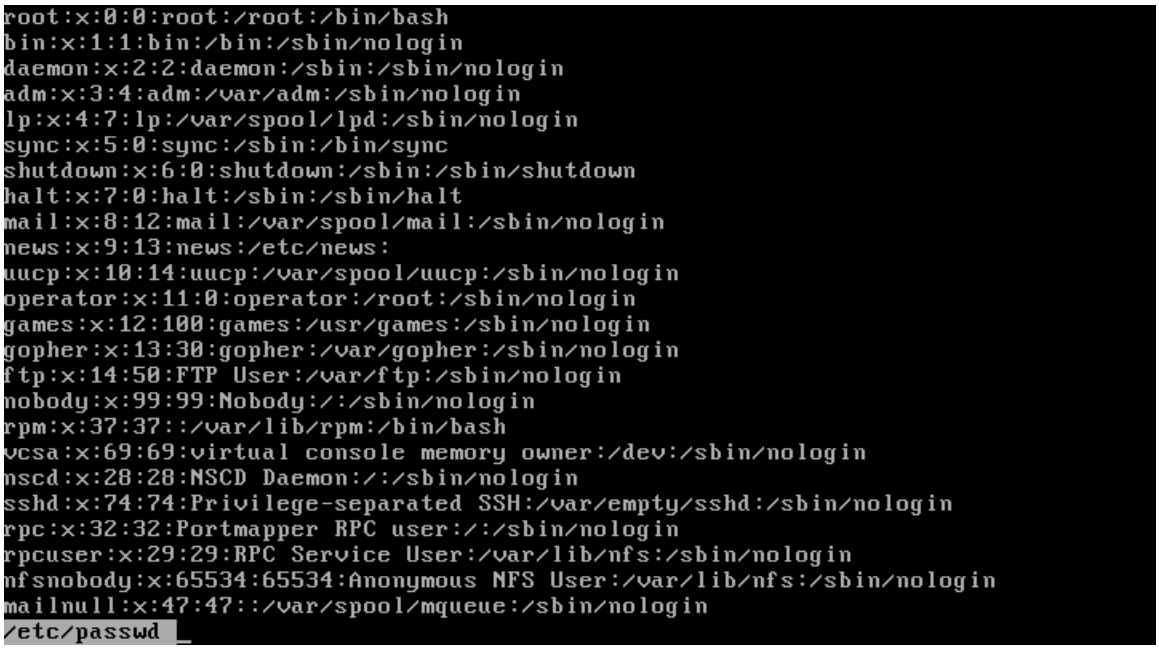
As you gain more Linux experience, you will find yourself telneting or ssh-ing to other systems on your network, or on the Internet. Knowing which account you are currently logged in as is vital; to determine this, type **whoami**.



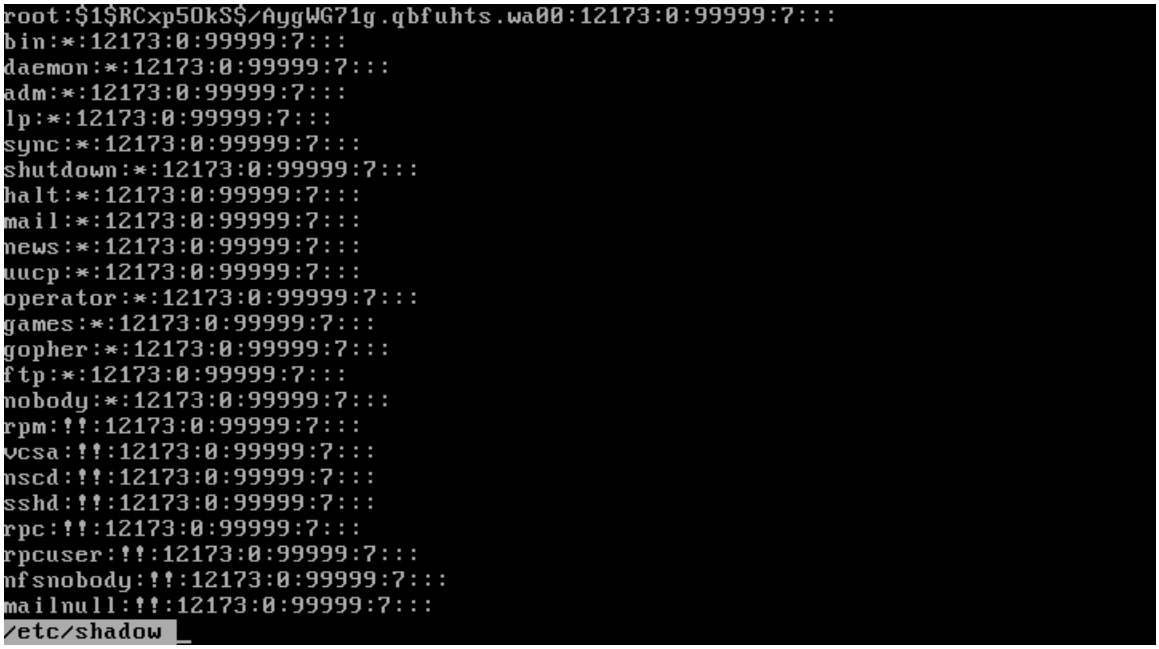
As shown in the previous screen, even if you initially logged on as a user and then **su**'ed to root, **whoami** will determine your current effective access level.

Common Files and Directories

Some user information is stored in a file called **passwd**, which is located in the **/etc** directory. This file also contains the path to the user's home directory, as well as to the current shell. Issue the command **less /etc/passwd** to view the contents of **passwd**. Press **q** to exit **less**.



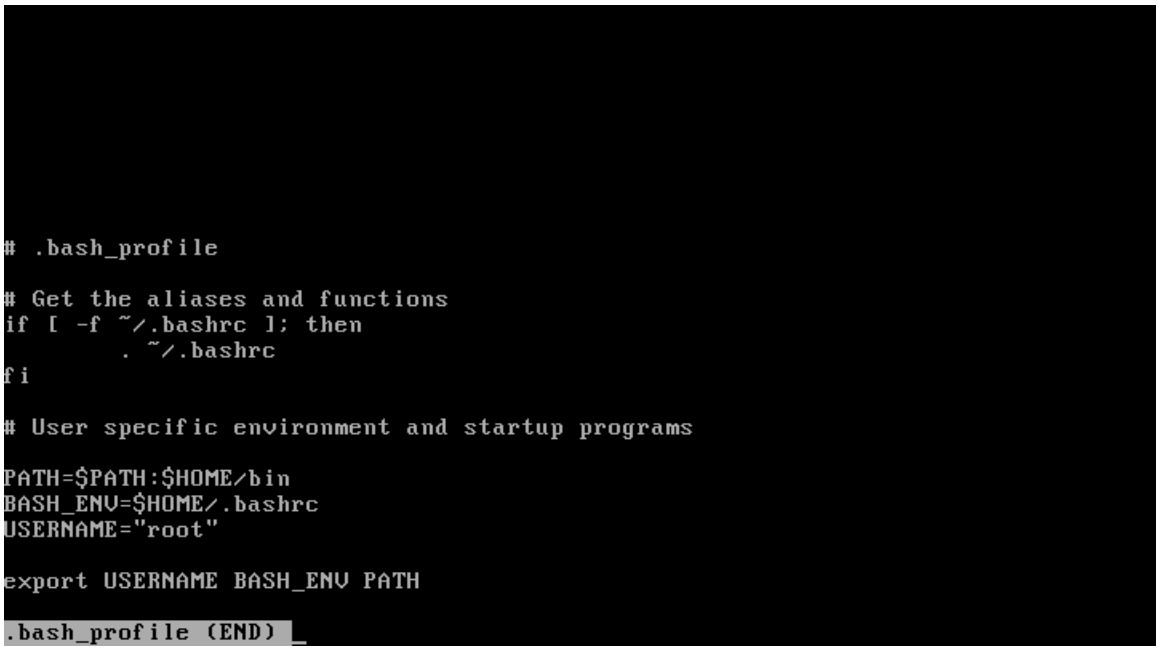
Notice that since the Red Hat default is to use a **shadow** file, the password for all accounts is listed as **x**. The **shadow** file contains an encrypted version of the actual password, and is used to enhance security. The permissions on the **shadow** file are generally more restrictive than the **passwd** file. To view the contents of the **shadow** file, issue the command **less /etc/shadow**.



Like Windows XP, Linux uses a **hosts** file that contains the IP address and associated hostname for a particular device. In a default install of Red Hat, the **hosts** file contains only one entry for localhost. The location of the **hosts** file in Linux is **/etc/hosts**.

As with most things in Linux, there are several ways to control access to your system. One is with TCP Wrappers, which intercepts and filters incoming requests. TCP Wrappers is installed when Red Hat is installed. The configuration files are located in **/etc** and are called **hosts.allow** and **hosts.deny**. The first file, **hosts.allow**, lists addresses or address ranges that can access your system; conversely, **hosts.deny** lists the addresses or address ranges that cannot access your system. The rules in **hosts.allow** take precedence over **hosts.deny**, so be careful what you add to **hosts.allow**.

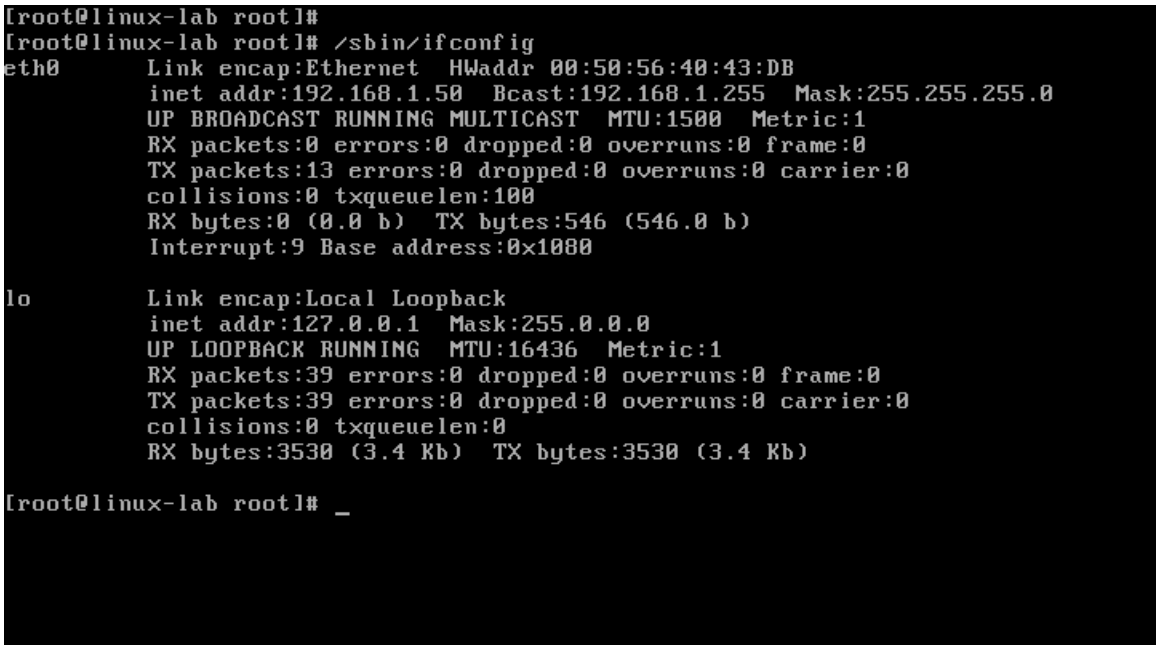
Some application installations require you to change the user's path. The path for a given user is specified in .**bash_profile**, which is located in the user's home directory. The default PATH statement can be viewed by issuing the command **less .bash_profile** while in the user's home directory, as shown in the following screen.



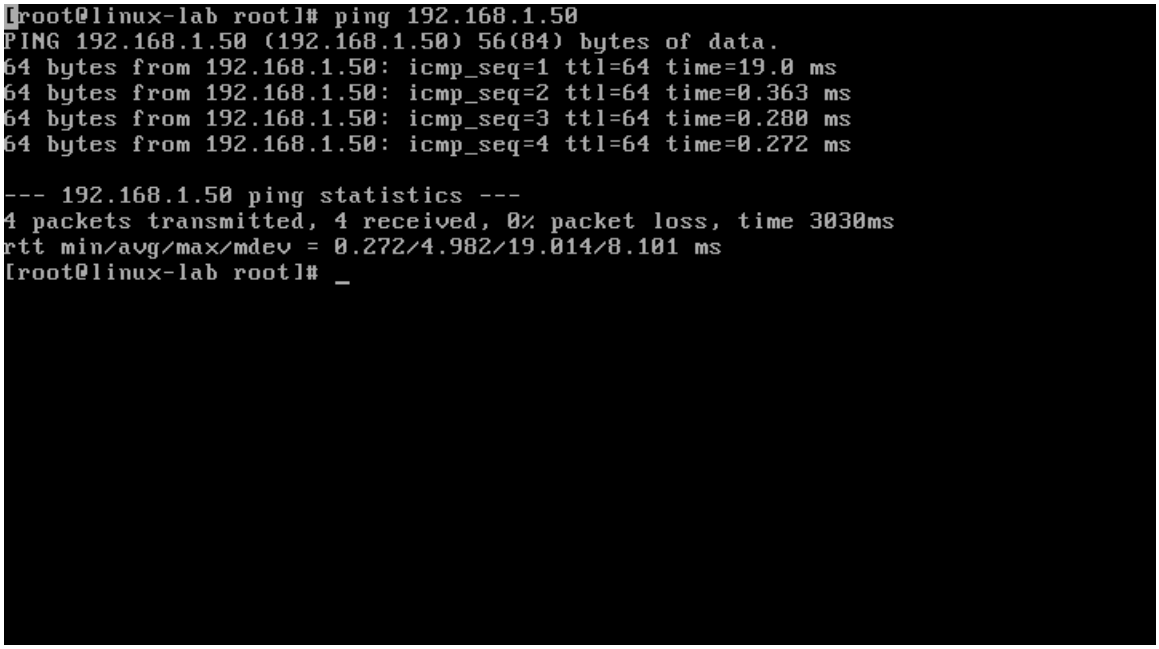
Many of the commands and applications we have discussed are front-ends for editing files that contain configuration information. So where are all of these seemingly important files? You will find the files that control network settings within the **/etc/sysconfig/networking** and **/etc/sysconfig/network-scripts** directory structures. The files that control common services such as echo, chargen, and time are located in **/etc/xinetd.d** and X windows configuration files are located in **/etc/X11**. Log files can be found in **/var/log**. Knowing the locations of these files is crucial when troubleshooting a problem or verifying that the system is running as required.

Network Configuration

Current network configurations can be viewed by issuing the **ifconfig** command. With Knoppix you should only receive an entry for 127.0.0.1 which is the loopback address.

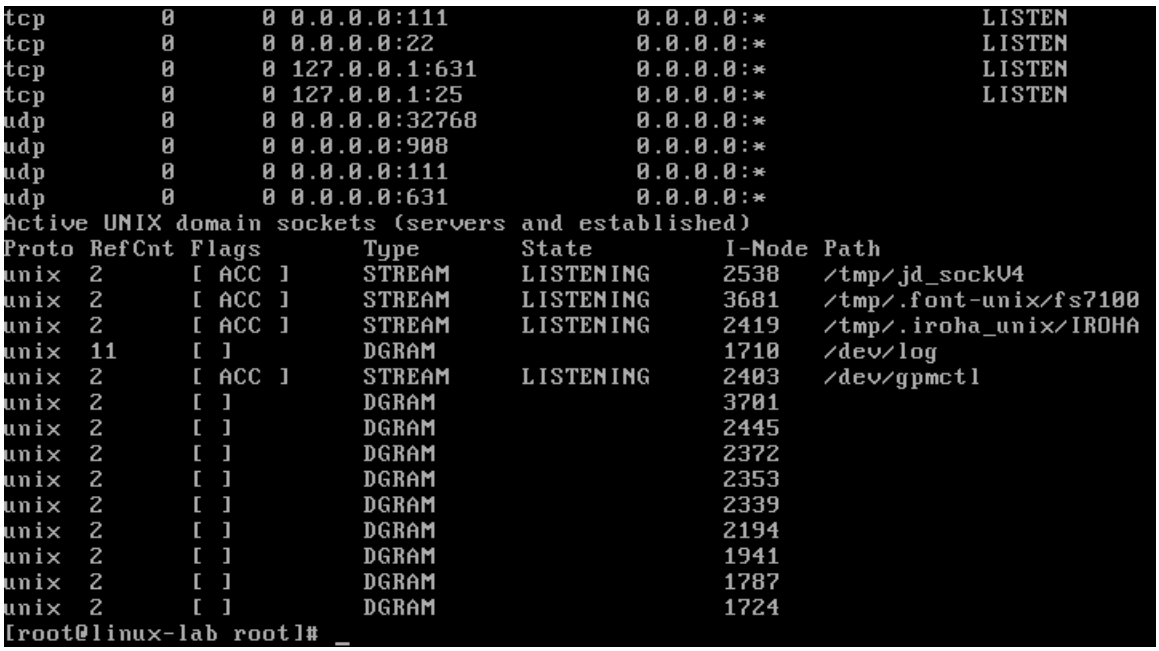


To verify that TCP/IP is setup correctly, simply **ping 127.0.0.1.** This will validate that the loopback is properly working. Pinging the loopback will be used for many of the exercises.



Listening Services and Network Connections

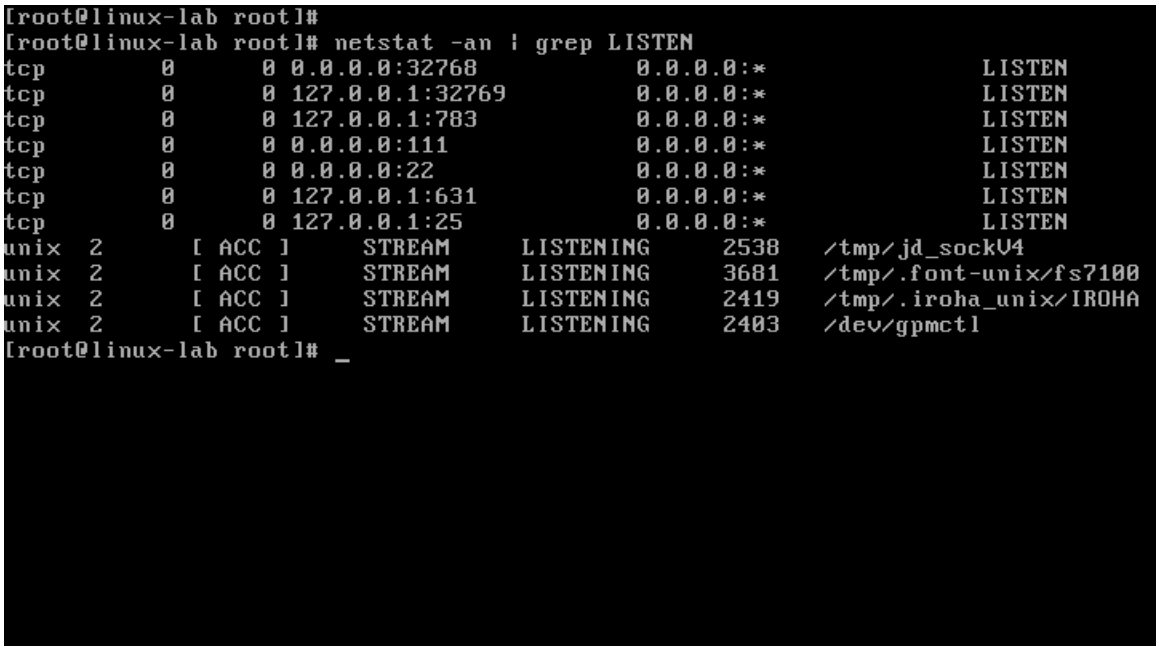
It is always important to know what services are listening on your system, as well as what connections have been made. Examples of listening services are **sendmail**, **rpc** and **sshd**; each of these listen on a specific port or a number of ports. To display the active network connections on your system, issue the **netstat** command.



```
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:32768           0.0.0.0:*
udp        0      0 0.0.0.0:908             0.0.0.0:*
udp        0      0 0.0.0.0:111             0.0.0.0:*
udp        0      0 0.0.0.0:631             0.0.0.0:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     2538   /tmp/jd_sockV4
unix  2      [ ACC ]     STREAM     LISTENING     3681   /tmp/.font-unix/fs7100
unix  2      [ ACC ]     STREAM     LISTENING     2419   /tmp/.iroha_unix/IROHA
unix  11     [ ]         DGRAM                    1710   /dev/log
unix  2      [ ACC ]     STREAM     LISTENING     2403   /dev/gpmctl
unix  2      [ ]         DGRAM                    3701
unix  2      [ ]         DGRAM                    2445
unix  2      [ ]         DGRAM                    2372
unix  2      [ ]         DGRAM                    2353
unix  2      [ ]         DGRAM                    2339
unix  2      [ ]         DGRAM                    2194
unix  2      [ ]         DGRAM                    1941
unix  2      [ ]         DGRAM                    1787
unix  2      [ ]         DGRAM                    1724
[root@linux-lab root]# _
```

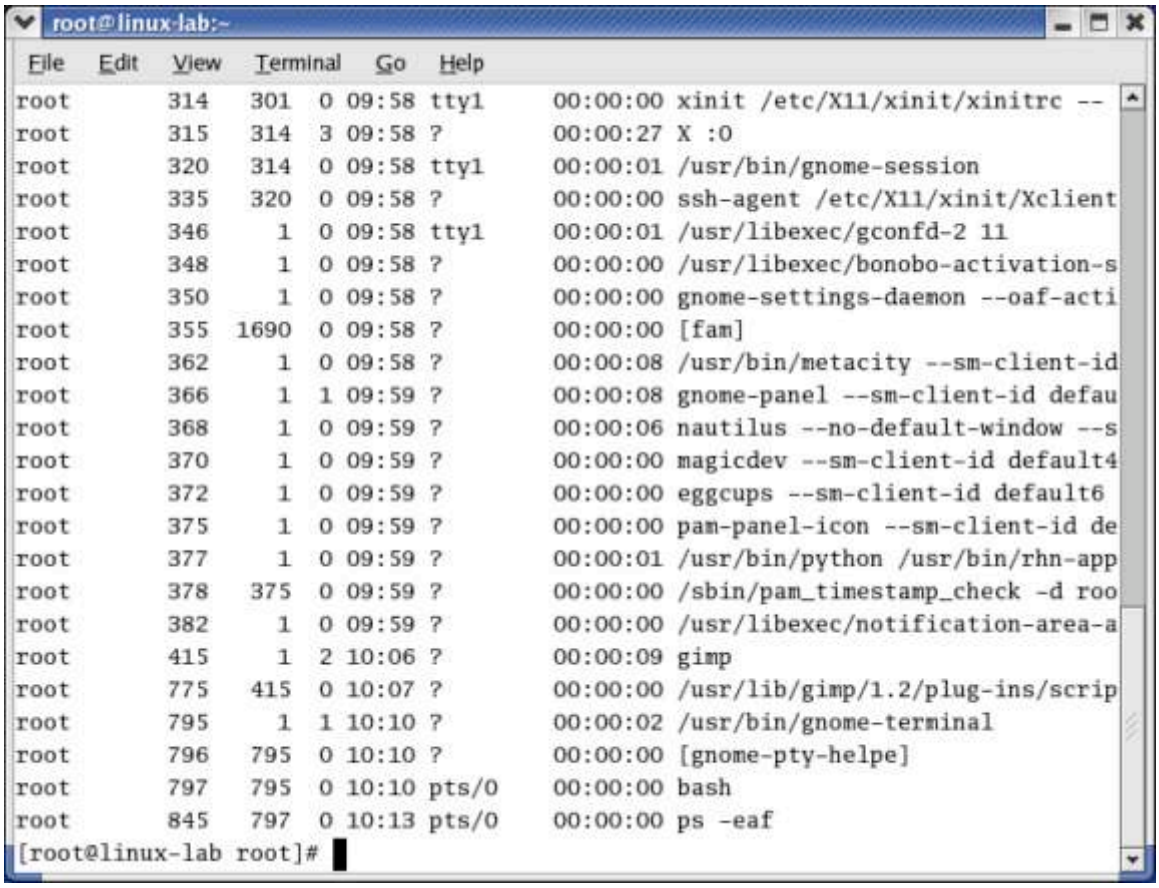
The **–an** option that was added to netstat specifies to list all connections (-a) and does not try to resolve hostnames (-n).

Depending on the applications you are running, the output may not fit onto one screen. This is a great time to pipe the output of one command through another. You can issue the command **netstat –an | more** to show one screen of information at a time, or you can use **grep** to search the output for specific requirements. The following screen shows that the command **netstat –an | grep LISTEN** outputs all of the servers that are listening on your system. Some of the other possible states besides **LISTEN** are **ESTABLISHED** and **TIME_WAIT**.



The ps Command

Linux gives the user the ability to run a command in the background by adding a blank space, and then **&** to the end of the command. To obtain a listing of currently running processes, including those that are running in the background, Linux provides the **ps** command. This command is invaluable for troubleshooting and for determining the current state of the system. Many options can be given to **ps** to control what it outputs to the command shell. For example, type **ps –eaf** into a command shell and press **Enter**, as in the following screen.



In the event that a program becomes unresponsive, you can forcibly end it. In order to do so, you first need to know what process id (PID) it is using. To determine this, issue the **ps –eaf** command and find the entry for the unresponsive program. The second column from the left contains the PID. Issue the command **kill <PID>** to kill the program.

You should now have a basic understanding of Knoppix. As is the case with anything in life, the best way to understand a topic is to practice. The information contained in this section should provide you with a basic knowledge to navigate the file system, perform basic configuration changes, and install applications. Many security tools written for Linux are not developed with the newbies in mind. Taking the time to learn the tools will provide a powerful, yet free toolbox for assessing the security of your network.