

Linux Terminal Services Setup

Typical Home LAN Topology
Linux-Mandrake Version 9.1

Arthur M. Royce (arthur.royce@verizon.net)

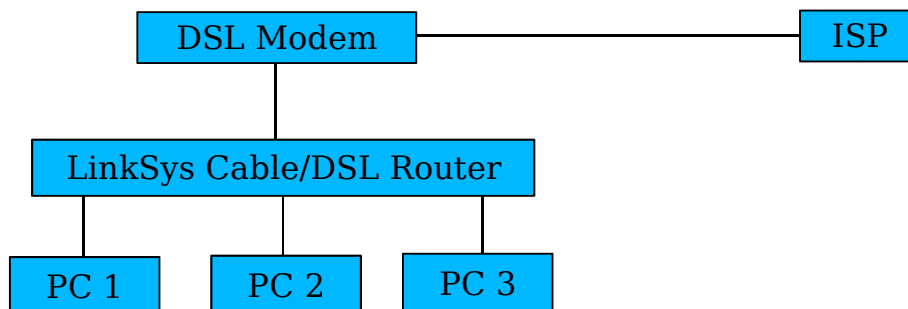
Introduction

After hearing about the Linux terminal services rolled-out in Largo, Florida, I was intrigued with this. This reminded me of my days performed Unix application development using color X Terminals networked into a Sun Sparc system. I have a 9 year old and an old computer that I would like to give her. We also have a network and a DSL connection to the internet. Since the machine is old it runs Linux adequately at best. My goal was to setup Linux terminal services so that my daughter could have a computer and so that I could control what programs she had access to and block her internet connection as well. My Linux computer has the following basic specifications:

- 2.0 Ghz Pentium 4 Processor
- 512 MBytes PC2100 DDR SDRAM
- 400 Mhz FSB
- ATI Rage 128 (32 Mbyte RAM)
- Integrated sound and network

Initial Setup

Looking at the sketch below I think this is how most homes and small businesses are setup. Everybody plugs their computer into the router where the router handles DNS, provides a basic network firewall and connects to the internet service provider. This is very simple and works quite well.



The first thing I did was to download all of the basic software from the Linux Terminal Services Project (<http://ltsp.org>) web site including the documentation. I read the documentation first. Next, I installed all of the software. I did not install any of the software to run programs on the client machine, just the server side components. I was purely interested in implementing basic terminal services.

Configuration Steps:

The first thing that I had to do was establish DHCP on my Linux machine. I disabled DHCP on the LinkSys router. I created a basic DHCP configuration file on my Linux machine which would simply hand out a set of addresses to any computer which requested one. I also established a route back to the LinkSys router. I realized that my Linux machine would have to boot with a static IP address since the DHCP daemon would not be started until after the eth0 interface was activated. I suppose I could change the order in which the DHCP service was activated, but I saw no problem creating a static IP address.

Linux server modifications:

Set static IP address (Linux Server): 192.168.1.50

Default Gateway (LinkSys Router): 192.168.1.1 (no change)

Added Additional Route on Linux machine:

Network: 192.168.1.0

Netmask: 255.255.255.0

Gateway: 192.168.1.1

When I type in route as root here is what is displayed:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	*	255.255.255.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	192.168.1.1	0.0.0.0	UG	0	0	0	eth0

I setup my DHCP configuration file initially just to hand out IP addresses. I tested this by booting the other computers on my network and ensured that they received the correct IP address based on the DHCP configuration file.

Now I was ready to configure the lts.conf, hosts and update the dhcp.conf file again. I also had to make a change to the tftp configuration file because my system was setup differently than the

default paths provide by the initialization script. If you are not running RedHat Linux then check this script. I also want to note that I setup remote sound on one of the machines after I had it booting as a Linux client. What kid doesn't need sound? As before, I installed the software per the instructions and then began the configuration updates in the lts.conf file. I left most of the files setup by the LTSP initialization script as intact as possible. Also, be sure to review the exports and hosts.allow files that are modified when you run the initialization script. I did not have to change mine.

I gained a lot of information looking at how the workstations were configured initially. For example, the Midwest Micro computer already had Linux installed, so I simply looked at the existing configuration files to determine how to setup the terminal services basic and sound configuration. It might be more of a challenge if you are upgrading a Windows computer to a Linux terminal. At a minimum you will need the MAC address of the NIC card. You can probably use the defaults at first and then adjust from there.

I went to the <http://www.Rom-O-Matic.com> web site to create boot floppies which matched each network card. I simply followed the directions and the boot floppies worked just fine. One issue that I did note is that Microsoft Windows seems to alter registers on the LinkSys 10/100 NIC. When I tried to boot off the floppy I received error messages. I had to completely turn the machine off and the boot floppy would work just fine. I confirmed this behavior for Windows 98 and Windows XP.

I obfuscated some of the information unique to my setup. Here is the tftp configuration file that I use:

```
service tftp
{
    disable                = no
    socket_type            = dgram
    protocol               = udp
    wait                   = yes
    user                   = root
    server                 = /usr/sbin/in.tftpd
    #server_args           = -s /var/lib/tftpboot
    # Modified for LTSP configuration for Mandrake Linux path
    server_args            = -s /tftpboot
    per_source             = 11
    cps                    = 100 2
    flags                  = IPv4
}
```

Here is the hosts file on my machine:

```
127.0.0.1      localhost
192.168.1.50   LocalLinuxBox.com
192.168.1.130  ws001
192.168.1.131  ws002
192.168.1.132  ws003
```

Here is the lts.conf file on my machine:

```
#
# Config file for the Linux Terminal Server Project (www.ltsp.org)
#

[Default]
SERVER      = 192.168.1.50
XSERVER     = auto
  X_MOUSE_PROTOCOL = "PS/2"
  X_MOUSE_DEVICE   = "/dev/psaux"
  X_MOUSE_RESOLUTION = 400
  X_MOUSE_BUTTONS  = 3
  USE_XFS          = N
  LOCAL_APPS      = N
  RUNLEVEL        = 5

# This section is for the computers that are being supported.
# Dell laptop
[ws001]
  XSERVER      = auto
  LOCAL_APPS   = N
  USE_NFS_SWAP = N
  SWAPFILE_SIZE = 64m
  RUNLEVEL     = 5
# HP workstation (I should configure the sound sometime)
[ws002]
  XSERVER      = auto
  LOCAL_APPS   = N
  USE_NFS_SWAP = N
  SWAPFILE_SIZE = 64m
  RUNLEVEL     = 5
# Old Midwest Micro computer
[ws003]
  # Added based on the sound configuration example.
  SOUND        = Y
  SOUND_DAEMON = nasd # This can be 'nasd', or 'esd' at this time
  VOLUME       = 100 # Speaker & WAVE volume percentage
  MIC_VOLUME   = 100 # Microphone volume
  CD_VOLUME    = 75  # CD Audio volume
  SMODULE_01   = soundcore
  # Set values based on current soundcard configuration.
  SMODULE_02   = ad1848 io=0x530 irq=10 dma=1
```

```
XSERVER          = auto
# This computer uses com port 1 for the mouse
X_MOUSE_PROTOCOL = "Intellimouse"
X_MOUSE_DEVICE   = "/dev/ttyS0"
LOCAL_APPS       = N
USE_NFS_SWAP     = N
SWAPFILE_SIZE    = 64m
RUNLEVEL         = 5
```

Here is the dhcp.conf file on my machine:

```
ddns-update-style none;
subnet 192.168.1.0 netmask 255.255.255.0 {
    # default gateway
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;

    # Local machine name - o.k. I am not very original.
    option domain-name "LocalLinuxBox.com";
    # ISP DNS servers
    option domain-name-servers xx.xx.xx.xx, xx.xx.xx.yy;

    # Set IP Address lease parameters
    # Keep these short so that the terminal workstations
    # can be powered down anytime and the address will release.
    default-lease-time 3600;
    max-lease-time 7200;

    # Specify dynamic IP address range
    range dynamic-bootp 192.168.1.100 192.168.1.125;

    # This section is for the LTSP setup.
    # LocalLinuxBox.com is the domain name.
    option root-path "192.168.1.50:/opt/ltsp/i386";

    group {
        use-host-decl-names on;

        # This is for my Dell i610 laptop
        host ws001 {
            # 3Com 3C920 (3C905c-TX compatible)
            hardware ethernet xx:xx:xx:xx:xx:xx;
            fixed-address 192.168.1.130;
            filename "/lts/vmlinuz-2.4.21-ltsp-1";
        }
        # Wifes HP workstation
        host ws002 {
            # LinkSys LNE100TX
            hardware ethernet xx:xx:xx:xx:xx:xx;
            fixed-address 192.168.1.131;
            filename "/lts/vmlinuz-2.4.21-ltsp-1";
        }
    }
}
```

```

    }
    # Clunky old Midwest Micro computer (daughter)
    host ws003 {
        # LinkSys LNE100TX
        hardware ethernet xx:xx:xx:xx:xx:xx;
        fixed-address      192.168.1.132;
        filename           "/lts/vmlinuz-2.4.21-ltsp-1";
    }
}

```

Now that everything was working fine, I wanted to add some firewall rules to prevent internet access. All I am concerned about right now is port 80 (http). I can always add more rules later if that is warranted. I made sure that IPTables was active and initiated the following commands as root:

```

iptables -A OUTPUT -m owner --uid-owner 'user id1' -p tcp --dport 80 -j REJECT
iptables -A OUTPUT -m owner --uid-owner 'user id2' -p tcp --dport 80 -j REJECT
iptables -A OUTPUT -m owner --uid-owner 'user id3' -p tcp --dport 80 -j REJECT

```

This effectively blocks all of my X clients from accessing the internet. I tested this from the client machines prior to adding this rule and internet access worked fine and did not seem to have much effect on the host server.

Summary

After installing the software on ltsp.org, I reconfigured my network by disabling DHCP on my router and modified the following files:

- tftp
- hosts
- dhcp.conf
- lts.conf

I also gave my server a static IP address and added a route in my server to my LinkSys router.

I validated the modifications automatically made to:

- hosts.allow
- exports

Make sure that you have dhcp and nfs (all components) running.

Create boot floppies appropriate for your network card(s). If you are booting from a Windows machine you might need to power-down the machine. The hardware reset does not seem to clear all the registers on the network card.

Issues:

1. Certain video-intensive games do not work adequately. It is the 100 Mbit LAN limitation, I believe. Terminal services is not for games, so I don't see this as a significant issue.
2. Booting from a floppy can get kind-of old. Creating an option ROM on the LAN card would be ideal.
3. For multiple printers I would setup a print server and configure it on the main server. Then printing is not an issue. These are pretty cheap and easy to use. Right now the printing works from my server to my laser printer connected to LPT1.

Opportunities:

1. All of the productivity applications that I have tested work just fine meaning that their performance is good.
2. An old computer becomes truly useful again.
3. No more licensing fees to Microsoft.
4. All administration is handled at the server level. Imagine solving a users problem by simply telling them to cycle power on their "appliance"?
5. The memory and performance "hit" on the server does not seem to be too great once the applications are running. If this were for a small office, I would load-up the RAM and want at least two hard drives instead of a single one. I would use one for the /boot and /home partition and another for the supporting paths /bin, /usr, /opt, etc. I would need to think this through, but you get the idea.
6. With three machines connected (this is all I have) the memory and processor impact seems minimal. With a single disk drive, application and file system access is what impacts my system.