

---

# Installing and Configuring the Linux Environment for ITLab PCs

Jianping Wang  
**University of Virginia**

Prepared for the ITL Workshop, June 2001

## Overview

This document describes a set of tasks for installing and customizing Linux. The exercises are intended for people with limited exposure to installing Linux or other Unix operating systems. The document is written to be usable as a reference.

## Disclaimers

The installation and configuration is intended for machines in an ITLab, and is not intended for machines connected to the Internet. The installation procedure described in this document will leave a system highly vulnerable to attacks.

If you use the information in this document for your own equipment, you will do so at your own risk. We are not responsible for any damage that may result from following the instructions or suggestions in this document.

This is the first revision of the document. We appreciate any feedback on the usefulness of the document and corrections of typos.

## Table of Contents

|  |    |
|--|----|
| Recommended Hardware .....             | 2  |
| Installing RedHat7.1 .....             | 2  |
| Recompiling the RedHat7.1 Kernel.....  | 6  |
| Customizing Linux.....                 | 8  |
| Installing software for the ITLab..... | 11 |

## Recommended Hardware

Hard disk space with at least 2GB space reserved for Linux;  
CD-Rom and floppy drive;  
Two Ethernet cards;  
Video card and Monitor;  
Keyboard and mouse;  
FDDI card (optional).

## Installing RedHat7.1

The following steps go through a RedHat7.1 Linux Installation from a CD-ROM. We assume that a copy of the RedHat7.1 CD-ROM distribution is available.

### Step 1: Preparing the Installation.

- Insert the disk 1 of RedHat7.1 CD-ROMs in the CD-ROM driver. Reboot the machine, press "Del" to enter into BIOS setup program. Select "CDROM" as the 1<sup>st</sup> boot device. Save the settings and exit.
- When the machine reboots, it will automatically enter the RedHat7.1 installation process. Follow the instructions on the screen to finish your customized installation step by step.
- Choose a the installation mode. In the **Installation Mode Selection** screen, select "text <ENTER>".

### Step 2: Starting the Installation.

- Choose a language. In the **Language Selection** dialog box, select English. .
- Selecting a keyboard type. In the **Keyboard Selection** dialog box, select U.S.-style keyboard.
- Select installation type. In the **Installation Type** dialog box, select "Custom System".

### Step 3: Creating Partitions.

- If you are installing Linux on a brand-new system, this is a good opportunity to allocate and create partitions for the Linux operating system.
- Select partition mode. In the **Automatic Partitioning** dialog box, choose "Manually partition".

- Select disk setup mode. In the **Disk Setup** dialog box, you can choose "Disk Druid" or "Fdisk", to view, add, delete and modify the partitions for the Linux system.
- Execute the partition. In the **Current Disk Partitions** dialog box, add, delete and edit partitions. Each partition typed "Linux native" should be arranged a mount point. One Linux partition with type "Linux native" and size of at least 600MB, should be added, and mounted on root directory "/". This partition will be used to install the system.

It is suggested to create a Linux swap partition. The size of the Linux swap partition depends on the amount of RAM on your machine, but not less than 16MB. It is recommended that

$$\text{Size of SWAP} = 2 \times (\text{Size of RAM})$$

Press "Ok" if the partitioning is finished.

- Execute the formatting. In the **Choose Partitions to Format** dialog box, select all "Linux native" partitions. Select "Check for bad blocks during format", but that will take a long time to format the partitions. Press "OK".

#### Step 4: Configuring LILO.

- In the **LILO Configuration** dialog box, press "OK".
- Choose the place where LILO is running. In the **LILO Configuration** dialog box (continuous), select "Master Boot Record (MBR)".
- In the **LILO Configuration** dialog box(continuous), check the LILO configuration. Edit it if necessary. Press "OK".

**Note:** The default configuration file for LILO is /etc/lilo.conf. You can manually update this file as root and execute the command "lilo" to make the new configuration permanent for the following reboots.

#### Step 5: Configuring Network.

- Configure network. In the **Network Configuration** dialog box, unselect "Use bootp/dhcp". Input IP address, netmask, default gateway (IP), primary nameserver (IP), second nameserver (IP) and ternary nameserver (IP). Press "OK".
- Input the hostname. In the **Hostname Configuration** dialog box, Input a hostname such as gold.
- Select the security level. In the **Firewall Configuration** dialog box, choose "No firewall" for "Security Level".

**Note:** The TCP/IP configuration can be changed manually after the installation is finished. There are several configuration files in RedHat7.1, which are used for saving the network configuration. Please refer to RedHat7.1 manuals for detailed information.

### **Step 6: Selecting Mouse, Language Support and Time Zone.**

- Choose the type of mouse. In the **Mouse Selection** dialog box, select "Generic – 3 Button Mouse (PS/2)".
- Choose the language support. In the **Language Support** dialog box, select "English (USA)".
- Choose time zone. In the **Time Zone Selection** dialog box, unselect "Hardware clock set to GMT?". Choose "America/New\_York". Press "OK".

### **Step 7: Inputting Root Password, Setting Up User Accounts and Configuring Authentication.**

- Input the root password. In the **Root Password** dialog box, input password twice.
- Add users. In the **Add User** dialog box, if you want to add new account, input user ID, user password and full name. Press "OK".
- Edit user accounts. In the **User Account Setup** dialog box, you can add, delete or edit user accounts here. Press "OK" if done.
- Choose password type. In the **Authentication Configuration** dialog box, select "Use Shadow Passwords". Unselect "Enable MD5 Passwords", "Enable NIS", "Enable LDAP" and "enable Kerberos". Press "OK".

### **Step 8: Selecting and Installing Packages.**

- Select the packages to be installed. In the **Package Group Selection** dialog box, select the components and packages you want to install. Selecting the required packages to install by specifying the individual packages is a tedious work. Usually you need to unselect "Select individual packages" checkbox and just choose the components you plan to install. If you are a Linux novice, it may be advisable to select all packages, unless a package has an explicit conflict with the configuration of your system. Press "OK" if done.
- In the **X probe results** dialog box, press "OK".
- In the **Installation to begin** dialog box, press "OK".
- A **Formatting** progress window pops up, showing the process that each selected partition is being formatted.
- A **Copying Files** progress window pops up, showing the progress that the install images are being transferred to hard driver.

- A **Package Installation** progress window pops up, showing the progress that all selected packages are being installed. Disc 2 is needed during the package installation process.

### **Step 9: Creating a Bootdisk.**

- We strongly recommend to create a boot diskette. When the operating system fails to boot after changing some configuration files or recompiling the kernel, the boot diskette can be used to boot the system, and correct any settings.
- In the **Bootdisk** dialog box, choose "Yes".
- In the **Bootdisk** dialog box (continuous), insert a blank floppy into the floppy driver. Press "OK".

### **Step 10: Configuring X Windows.**

- In the **Monitor Probe** dialog box: Check the parameters of the monitor detected by the system. Choose "Yes" if it's correct. Otherwise you need to choose "No" to configure the monitor manually.
- In the **Probing to begin** dialog box, press "OK".
- In the **Probing finished** dialog box, choose "Use Default" if the probed result is correct.
- In the **Starting X** dialog box, choose "Ok". Choose "Yes" in the following two questions.

### **Step 11: Completing the Installation.**

- In the **Complete** dialog box: Press "OK".
- The installation is completed and the machine is rebooted automatically.

## Recompiling the RedHat7.1 Kernel

The installation program described above creates a kernel with default settings. For the ITLab, the Linux installation needs to be customized, e.g., add FDDI card support, turn on IP forwarding. A customization of the kernel requires that the kernel be recompiled.

For customized installations, the RedHat7.1 kernel source code package is needed.

You can download the kernel source code package (named linux-2.4.5.tar.gz) from the website: <http://www.kernel.org/pub/linux/kernel/v2.4/>. The following steps describe how to compile the kernel using linux-2.4.5.tar.gz.

**Step 1:** Download the source codes package of kernel 2.4.5, linux-2.4.5.tar.gz from <http://www.kernel.org/pub/linux/kernel/v2.4/> and put it on the directory /usr/src.

**Step 2:** Go to /usr/src and install the software with the following command:

```
tar -zxvf linux-2.4.5.tar.gz
```

A subdirectory /usr/src/linux will show up after installing the kernel package.

**Step 3:** cd /usr/src/linux/, type the command:

```
make menuconfig
```

when the kernel configuration menu is popped up, select the required items:

```
main-menu → Networking options →
[*] IP: advanced router
[*] IP: verbose route monitoring (NEW)
[*] IP: multicast routing
main-menu → Networking device support → Ethernet (10 or
100Mbit) →
<*> PCI NE2000 and clones support (see help)
main-menu → Networking device support →
[*] FDDI driver support
<*> Digital DEFEA and DEFFA adapter support (NEW)
<M> PPP (point-to-point protocol) support
<M> SLIP (serial line) support
[*] Fibre Channel driver support
main-menu → File systems →
<*> DOS FAT fs support
<*> MSDOS fs support
<*> VFAT (Windows 95) fs support
main-menu → Console drivers →
[*] Video mode selection support
```

Return to the main menu. Select exit to save the selected items and exit the configuration program.

**Step 4:** In /usr/src/linux, execute the following commands:

```
make dep
make clean
```

### **make bzlilo**

**Step 5:** If you selected some items as modules, for instance:

```
<M> PPP (point-to-point) support
<M> SLIP (serial line) support
```

execute the following commands:

```
make modules; make modules_install
```

**Step 6:** If everything goes well, a new kernel, named `vmlinuz`, will be created and copied to the directory `.`.

**Step 7:** If you want to run the new kernel when rebooting the machine, you need to update

`/etc/lilo.conf`. Replace the following lines

```
image=/boot/vmlinuz-2.4.2-2
    label=linux
    read-only
    root=/dev/hda1
```

with the lines

```
image=/vmlinuz
    label=linux
    read-only
    root=/dev/hda1
```

After updating the file, `/etc/lilo.conf`, execute the command  
**lilo**

**Step 8:** Reboot the machine.

## Customizing Linux

In order for the RedHat7.1 system to have the required features, the system often needs to be reconfigured. This section describes various customization of the system that may be need for the ITL Labs.

### 1. Set up IP forwarding.

IP forwarding must be enabled whenever a PC has to act as a router. The file `/etc/sysconfig/network` specifies whether a machine supports IP forwarding or not. If the machine acts as a router, this file includes a line:

```
FORWARD_IPV4=yes
```

If the machine is a host, the line does not exist or is set to `FORWARD_IPV4=no`.

Unfortunately, according to our experience, there is no system script file which reads and deals with the above configuration line during the initialization phase of the system. To ensure the system can process the configuration of IP forwarding function correctly, the following lines are added to the end of the script file, `/etc/rc.d/rc.local`.

```
. /etc/sysconfig/network
if [ -d /proc/sys/net/ipv4 ];
then
    if [ ${FORWARD_IPV4} = "yes" -o ${FORWARD_IPV4} = "true" ];
    then
        echo 1 > /proc/sys/net/ipv4/ip_forward
    else
        echo 0 > /proc/sys/net/ipv4/ip_forward
    fi
fi
```

### 2. Set up a permanent IP address and a startup static routing table.

In RedHat7.1, suppose you have two Ethernet interfaces on the machine, there are two files which contain the network configurations of these Ethernet interfaces:

```
/etc/sysconfig/network-scripts/ifcfg-eth0    //for interface eth0
/etc/sysconfig/network-scripts/ifcfg-eth1    //for interface eth1
```

The contents of `ifcfg-eth0` may read like this:

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=10.0.14.255
IPADDR=10.0.14.2
NETMASK=255.255.255.0
NETWORK=10.0.14.0
ONBOOT=yes
```

If the content of this file is modified, the updates will take effect during the next reboot of the machine.

The start-up static routing table is specified in the file:

`/etc/sysconfig/static-routes`,  
whose contents may be:

```
eth0 net 10.0.1.0 netmask 255.255.255.0 gw 10.0.13.4
eth0 net 10.0.6.0 netmask 255.255.255.0 gw 10.0.1.3
eth1 net 10.0.7.0 netmask 255.255.255.0 gw 10.0.13.4
eth0 default gw 10.0.11.1
```

When the system is booted, each line of the file `static-routes` is run as the parameters of the command "route".

### 3. Change the hostname

In RedHat7.1, there is no "easy" way to change the hostname. The command, `hostname host-name`, just changes the hostname until the next reboot of the system. To permanently change the hostname, you need to edit the following files manually:

```
/etc/HOSTNAME
/etc/sysconfig/network
/etc/hosts //if necessary
/etc/resolv.conf //if necessary
```

### 4. The boot manager Lilo

Lilo is a program which can send/receive information to/from the boot sectors. The configuration file of Lilo, `/etc/lilo.conf`, contains the data Lilo uses. Below is a typical example of `lilo.conf`:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=5000
linear
default=linux

image=/boot/vmlinuz-2.2.14-5.0
    label=linux
    initrd=/boot/initrd-2.2.14-5.0.img
    read-only
    root=/dev/sda3
other=/dev/hda1
    label=bsd
```

The above setting is for a system with two operating systems, so-called dual-boot system. Here, Lilo lets you boot the computer with Linux (with boot label `linux`) or FreeBSD (with boot label `bsd`).

When the machine is booted, the prompt `lilo`, appears. You select one of the operating systems, by typing the label name followed by the Enter key. Pressing the "Tab" key displays all label names managed by lilo.

If you make changes to `/etc/lilo.conf`, such as recompiling the kernel or changing the timeout value, you need to execute the command "lilo" as root to update the boot sector (MBR). The updates will be effective at the next reboot of the machine.

## 5. X Windows

Sometimes you need to reconfigure X Windows, e.g., when replacing the monitor or the video card. An X configuration program, `Xconfigurator`, can be used to carry out the reconfiguration of X. When `Xconfigurator` is running, several configuration dialog boxes appear one by one. Follow the instructions in the dialog box to select the appropriate type and parameters. For video settings, we recommend 24-bit color and a resolution of 1024x768.

## 6. ICMP redirect

ICMP redirect packets are used by routers to inform a host about a better route when the host uses a non-optimal or defunct route to a particular destination. Since forged ICMP redirect packets can be used by a malicious attacker to alter the routing tables on a host, most experts recommend to disable the ICMP redirect function on a router.

We have an experiment in Lab3 designed for the students to investigate how the ICMP redirect function works on a router. To enable support of ICMP redirect, add the following lines to file `/etc/sysctl.conf`:

```
# Enable ICMP Redirect Acceptance
net.ipv4.conf.all.accept_redirects = 1
```

and add the following commands at the end of the file `/etc/rc.d/rc.local`:

```
for f in /proc/sys/net/ipv4/conf/*/accept_redirects
do
    echo 1 > $f
done
```

In Linux, the host which receives ICMP redirect packets doesn't update its routing table, instead it changes the routing cache, which can be viewed by command "route -Cn".

## Installing software for the ITLab

There are several network tools or software used in the lab manual for the ITLab. All of them can be found on the Internet, and downloaded at no cost. We have downloaded the software required to run ITL Labs, and made it available at <http://www.cs.virginia.edu/~itlab/software>.

Below is a summary of the software packages, with summaries of the installation procedures.

Installing `inetd`, `in.telnetd`, `in.ftpd`, `in.fingerd`, `in.talkd`, `in.ntalkd`, `in.rshd`, `in.rlogind`

The `inetd` server, also called superserver, is a network daemon program that manages incoming network connections. It monitors all network interfaces. The configuration file of `inetd` has information on the program to run when an incoming connection is received. For example, when receiving a telnet connection, `inetd` calls `in.telnetd` to deal with the request. As a exercise, you will install them manually.

**Note:** For security reasons, you should close more network servers as possible, such as `ftp`, `telnet`, `shell`, `login`, `exec`, `talk`, `ntalk`, `imap`, `pop-2`, `pop-3`, `finger`, `auth`, etc. If a machine acts as a `www` and `ftp` server, only port 80, port 20 and port 21 need to be used for `www` and `ftp` services. All other unrelated ports should be closed.

- You can check if `inetd` is running on your system by typing:

```
ps -aux | grep "inetd"
```

- If `inetd` is not installed, download the following packages from the website <http://www.cs.virginia.edu/~itlab/software>. All of them are in `.rpm` format:

```
inetd-0.16-4.i386.rpm
telnet-server-0.16-6.i386.rpm
wu-ftpd-2.6.0-3.i386.rpm
finger-server-0.16-5.i386.rpm
talk-server-0.16-4.i386.rpm
rsh-server-0.16-12.i386.rpm
```

- To install a package in `rpm` format, execute the following command:

```
rpm -ivh inetd-0.16-4.i386.rpm
```

To uninstall the above `rpm` format package, type:

```
rpm -e inetd
```

- Customize the configuration file `/etc/inetd.conf`. Below is the contents of `/etc/inetd.conf`. The bold lines indicate important network services.

```

#
# inetd.conf      This file describes the services that will be available
#                through the INETD TCP/IP super server.  To re-configure
#                the running INETD process, edit this file, then send the
#                INETD process a SIGHUP signal.
# Version:       @(#)/etc/inetd.conf      3.10      05/27/93
#
# Authors:       Original taken from BSD UNIX 4.3/TAHOE.
#                Fred N. van Kempen, <waltje@uwalt.nl.mugnet.org>
#
# Modified for Debian Linux by Ian A. Murdock <imurdock@shell.portal.com>
# Modified for RHS Linux by Marc Ewing <marc@redhat.com>
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
# Echo, discard, daytime, and chargen are used primarily for testing.
#
# To re-read this file after changes, just do a 'killall -HUP inetd'
#
echo      stream  tcp      nowait  root    internal
echo      dgram  udp      wait    root    internal
discard   stream  tcp      nowait  root    internal
discard   dgram  udp      wait    root    internal
daytime   stream  tcp      nowait  root    internal
daytime   dgram  udp      wait    root    internal
#chargen  stream  tcp      nowait  root    internal
#chargen   dgram  udp      wait    root    internal
time      stream  tcp      nowait  root    internal
time      dgram  udp      wait    root    internal
#
# These are standard services.
#
ftp       stream  tcp      nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
telnet    stream  tcp      nowait  root    /usr/sbin/tcpd  in.telnetd
#
# Shell, login, exec, comsat and talk are BSD protocols.
#
shell     stream  tcp      nowait  root    /usr/sbin/tcpd  in.rshd
login     stream  tcp      nowait  root    /usr/sbin/tcpd  in.rlogind
#exec     stream  tcp      nowait  root    /usr/sbin/tcpd  in.rexecd
#comsat    dgram  udp      wait    root    /usr/sbin/tcpd  in.comsat
talk      dgram  udp      wait    nobody.tty  /usr/sbin/tcpd  in.talkd
ntalk     dgram  udp      wait    nobody.tty  /usr/sbin/tcpd  in.ntalkd
#dtalk    stream  tcp      wait    nobody.tty  /usr/sbin/tcpd  in.dtalkd
#
# Pop and imap mail services et al
#
#pop-2     stream  tcp      nowait  root    /usr/sbin/tcpd  ipop2d
#pop-3     stream  tcp      nowait  root    /usr/sbin/tcpd  ipop3d
#imap      stream  tcp      nowait  root    /usr/sbin/tcpd  imapd
#
# The Internet UUCP service.
#
#uucp      stream  tcp      nowait  uucp    /usr/sbin/tcpd  /usr/lib/uucp/uucico  -l
#
# Tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers." Do not uncomment
# this unless you *need* it.
#
#tftp      dgram  udp      wait    root    /usr/sbin/tcpd  in.tftpd
#bootps    dgram  udp      wait    root    /usr/sbin/tcpd  bootpd
#
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers." Many sites choose to disable
# some or all of these services to improve security.
#
finger    stream  tcp      nowait  nobody  /usr/sbin/tcpd  in.fingerd
#cfinger   stream  tcp      nowait  root    /usr/sbin/tcpd  in.cfingerd

```

```

#sysstat stream tcp nowait guest /usr/sbin/tcpd /bin/ps -auwx
#netstat stream tcp nowait guest /usr/sbin/tcpd /bin/netstat -f
inet
#
# Authentication
#
# identd is run standalone now
#
#auth stream tcp wait root /usr/sbin/in.identd in.identd -e -o
#
# End of inetd.conf

linuxconf stream tcp wait root /bin/linuxconf linuxconf --http

```

- The configuration file `/etc/services` defines the mapping between port number and standard network services. Make sure the file exists.
- If you want to allow a remote user to `telnet` or `ftp` into the local machine as `root`, you need to edit the following configuration files:

- `/etc/hosts.allow`

The file specifies which hosts are allowed to access `telnet` and `ftp` services on local machine. Set the file to only contain comments and blank lines (default).

- `/etc/hosts.deny`

The file specifies which hosts are disallowed to access to `telnet` and `ftp` services on local machine. Set the file to only contain comments and blank lines (default).

- `/etc/securetty`

The file lists all names of `tty` devices from which `root` can log into the system. Rename (for a backup file) or remove this file.

- `/etc/ftphosts`

This file is used to define whether users are allowed to log in from certain hosts or whether access is denied. Set the file to only contain comments and blank lines (default).

- `/etc/ftpusers`

The `/etc/ftpusers` file specifies those users that are **not** allowed to connect to your `FTP` server. Below is the contents of `/etc/ftpusers`:

```

root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody

```

Delete the line which contains "root" to allow a remote user to `ftp` to the local machine as `root`.

- The configuration file `/root/.rhosts` is used for running remote commands on remote commands using `rsh`. It contains host names and user names. The file allows a user on a machine with the IP address or hostname listed, to run `rsh` to execute commands remotely. In the ITLab, root can run `rsh` on all remote IP addresses. So, the file `/root/.rhosts` looks like this:

```
10.0.12.1 root
10.0.12.2 root
10.0.14.1 root
10.0.14.2 root
10.0.23.1 root
10.0.23.2 root
10.0.45.1 root
10.0.45.2 root
10.0.56.1 root
10.0.56.2 root
10.0.36.1 root
10.0.36.2 root
vintpc1 root
vintpc2 root
vintpc3 root
vintpc4 root
vintpc5 root
vintpc6 root
```

## Installing tcpdump

Tcpdump, a powerful network monitoring and analyzing tool, is widely used in the ITLab manual. The following steps install `tcpdump` in `/usr/local/bin`.

- Download the software `tcpdump_3.4a6.orig.tar.gz` and `libpcap.tar.gz` from the following website:

```
http://www.cs.virginia.edu/~itlab/software
```

- Install `libpcap` library and `tcpdump`.
  - Unzip them in some directory, for example, `/usr/src/tcpdump-linux`

- First install `libpcap`:

```
cd /usr/src/tcpdump-linux/libpcap-0.4
./configure
make
make install;make install-incl;make install-man
```

- Then install `tcpdump`:

```
cd /usr/src/tcpdump-linux/tcpdump-3.4a6.orig
(in Makefile.in, set BINDEST=/usr/local/bin,
MANDEST=/usr/local/man)
./configure
make
make install;make install-man
```

## Installing ethereal

- Download the ethereal software package `ethereal-0.8.14.tar.gz` from the following website to the installation directory, for example, `/root/ethereal-install`:

```
http://www.cs.virginia.edu/~itlab/software
```

- Install ethereal as follows:

```
cd /root/ethereal_install
tar -zxvf ethereal-0.8.14.tar.gz
cd /root/ethereal_install/ethereal-0.8.14
./configure
make
make install
```

- This creates two executable programs:

```
/usr/local/bin/ethereal
/usr/local/bin/tethereal
```

## Installing ttcp

- Download the ttcp software package, `ttcp-1.4-1.src.rpm`, from <http://www.cs.virginia.edu/~itlab/software> to the directory `/root/ttcp-install`.

- Install ttcp as follows:

```
cd /root/ttcp-install
rpm -ivh ttcp-1.4-1.src.rpm
    // (This will create a file, ttcp.shar, in the
    // directory /usr/src/redhat/SOURCES)
cd /usr/src/redhat/SOURCES
mkdir ttcp
mv ttcp.shar ttcp
cd ttcp
unshar ttcp.shar
gcc -o ttcp ttcp.c
cp ttcp /usr/local/bin
cp ttcp.lm /usr/local/man/man1
```

## Installing Kermit

- Download the kermit software package, `C-Kermit-7.0_196-1_i386.rpm`, from one of the following two websites to `/root/Kermit-install`.

```
http://www.cs.virginia.edu/~itlab/software
```

- Install kermit.

```
cd /root/kermit-install
rpm -ivh C-Kermit-7.0_196-1_i386.rpm
```

- It will create an executable program, `/usr/bin/kermit`, and a man file, `/usr/man/man1/kermit.1`

### Installing gated (RedHat 6.2 only)

- Download the gated software package, `gated-3-6.tar.gz`, from the following website to `/root/gated-install`.

`http://www.cs.virginia.edu/~itlab`

- Install gated as follows:

```
cd /root/gated-install
tar -zxvf gated-3-6.tar.gz
cd /root/gated-install/gated-public-3_6
./configure
make depend
make
cp root/gated-install/gated-public-3_6/gated-3-6/gated-
public-3_6/src/gdc/gdc /usr/local/bin
cp root/gated-install/gated-public-3_6/gated-3-6/gated-
public-3_6/src/ospf_monitor/ospf_monitor
/usr/local/bin
```

- The installation creates a set of executable programs: `gated`, `gdc`, `ospf-monitor` and `ripquery` in the directory `/usr/local/bin`.