

ttylinux Version 8.0

User Guide

Douglas Jerome
(Based on Previous Work by Pascal Schmidt)

December 14, 2008

Contents

1	Introduction	4
1.1	What is ttylinux?	4
1.2	What is ttylinux not?	4
1.3	What can ttylinux be used for?	4
1.4	Licenses	5
2	Starting with ttylinux	6
2.1	System Requirements	6
2.1.1	Custom Kernel Requirements	7
2.2	File Downloads	8
2.3	Booting the CD-ROM Image	8
2.4	Making a Bootable Flash or USB Disk from the ttylinux CD-ROM	9
3	Installation Guide	9
3.1	Installing from CD-ROM	9
3.1.1	Source CD-ROM Device	10
3.1.2	Target Partition Device	10
3.1.3	Boot Loader Location	12
3.1.4	Running the Installer	12
3.2	Manual Installation	13
3.3	Customization	14
4	Add-ons	16
5	System Guide	16

5.1	Basic Features	16
5.2	Bootup, Shutdown and System Configuration	17
5.3	Shell Environment	18
5.4	Using the Dropbear for SSH	18
5.5	Using an Ethernet Network	19
5.6	Using the Firewall	20
5.7	Using Dialup	20
5.8	Package Management	22
5.9	Using the sysconfig Script	23
5.10	Depricated and Legacy Items	24
5.10.1	devfs	24
6	Contact and Help	24
A	ttylinux-specific Commands Overview	24

1 Introduction

This section gives an overview of ttylinux, what it is and is not, and what it it might be used for.

1.1 What is ttylinux?

ttylinux is a small, minimal Linux distribution. It is available as a bootable CD-ROM image and an entire source distribution that builds the bootable CD-ROM image.

ttylinux tries to have as much as possible in as little space as possible. It provides multi-tasking, multi-user Linux with networking capabilities in 8 megabytes of disk space. It is prepared for Internet access by Ethernet and dial-up via modem or ISDN. A text-based web browser and command-line remote login secure client and server are a part of ttylinux.

ttylinux can be manually installed onto a hard disk or USB flash drive, or by using an installer script that is included on the CD-ROM. Both installation methods, automated script or manual, can be done by ttylinux or a different Linux system. However, ttylinux can provide only the lilo boot loader.

The ttylinux file system, excluding the Linux kernel, is 8 MB in size. Using an installer script provided on the bootable CD-ROM, a Linux kernel of about 2.2 MB will also be installed. This makes a minimum workable size of about 12 MB for a hard disk partition to install ttylinux on.

ttylinux includes a package management script capable of installing, removing and querying software packages and their files. This is useful for adapting ttylinux to specific needs.

1.2 What is ttylinux not?

ttylinux is not a typical Linux distribution; it does not have a graphical user interface, software development tools, document preparation and printing tools, databases nor network services such as BIND, News Server, Mail Transfer Agents and file servers.

1.3 What can ttylinux be used for?

ttylinux is useful on computers which are considered obsolete, such as 486SX PC; it is for people who want to have a minimal Linux distribution to run when little space is available or needed.

On a modern computer, one interesting use for ttylinux is as a tool for putting together a more complete Linux system.

With its small size `ttylinux` boots quickly from CR-ROM and USB flash drives; it has been used as a system fix/repair tool, and is a good basis for a rescue or installation CD-ROM.

`ttylinux` provides a working Linux environment with its boot image, and custom task-specific scripts can mount other parts of the file system from its boot USB flash drive or CD-ROM to provide a larger system.

Some users may want to use the `ttylinux` file system and configure and build their own Linux kernel.

`ttylinux` can serve as a rough prototype of a larger system, since it uses the same C library, `glibc`, as full Linux systems, compiling programs on a different Linux computer and copying them over to the `ttylinux` file system can result in working programs. **This is not a supported feature.** Programs compiled outside the normal `ttylinux` build process may require libraries not present in `ttylinux`. Worse, they may be compiled on a computer with different Linux kernel capabilities and make system calls not present in `ttylinux`.

Note: `ttylinux` is for people who have Linux experience; it is not for beginners, unless you want to learn how a Linux system works underneath the Graphical User Interfaces. You must be able to use the interactive shell command-line, and it helps to know your way around Linux system. Most of the programs are smaller versions of the common Unix utilities.

1.4 Licenses

The software packages that are part of `ttylinux` are licensed under a number of different open source licenses, as listed below. The initialization and system service scripts developed by the `ttylinux` project are licensed under the GNU General Public License; a copy of this license is included in the file `COPYING`.

Package	Version	License
<code>bash</code>	3.2	GPL
<code>busybox</code>	1.13.1	GPL
<code>dropbear</code>	0.51	MIT
<code>e2fsprogs</code>	1.41.3	GPL
<code>glibc</code>	2.7	LGPL
<code>iptables</code>	1.4.2	GPL
<code>isdn4k-utils</code>	3.2p1	GPL
<code>lilo</code>	22.8	BSD
<code>ncurses</code>	5.6	MIT
<code>ppp</code>	2.4.4	BSD
<code>retawq</code>	0.2.6c	GPL

For more information on the licenses, please visit the opensource.org website.

2 Starting with ttylinux

This section has a general overview of the ttylinux download image and also describes the system hardware requirements for using ttylinux, where to download ttylinux from, what to download and how to use the downloaded images.

ttylinux has three basic parts: a boot loader, a Linux kernel, and a root file system. All three of these are in the CD-ROM image; the CD-ROM image can be burned onto a blank CD-ROM disc and then booted. When booted, the root file system from the CD-ROM is decompressed and becomes a read/write root file system in a RAM disk in memory. Note that changes to any of the files while running ttylinux are lost, as they are in a RAM disk. Booting the ttylinux CD-ROM is described in section 2.3.

Installing ttylinux from the bootable CD-ROM onto a disk is described in section 3. Installation makes a system different from the bootable CD-ROM; the installed ttylinux has a read/write root file system directly on hard disk or flash disk, not in a RAM disk. The advantage of an installed ttylinux system over the RAM disk system is that file changes are not lost.

ttylinux can be put onto a flash disk, such as a USB drive, which can be made bootable. This copies the RAM disk boot method to the flash disk; when the flash disk is booted, the root file system from the flash disk is decompressed and becomes a read/write root file system in a RAM disk in memory. Changes to files are lost when the system shuts down. The process of putting ttylinux onto a flash disk described in more detail in section 2.4.

The ttylinux root file system is a file on the CD-ROM; it can be copied and used with a different custom kernel, one that you make, and put onto other media, or file system image, with your boot loader of choice. This process is beyond the scope of this document, but the requirements for a ttylinux custom kernel are described in more detail in section 2.1.1.

2.1 System Requirements

ttylinux is intended to eventually run on several different CPU architectures; however, version 8 directly supports only the Linux i386 CPU architecture, this is the x86 architecture for PC compatible computers, specifically the i486 CPU instruction set.

CPU and Computer

ttylinux 8 requires i486SX or newer processor in a PC compatible computer. It will not work with the i386 CPU; the glibc version in ttylinux uses CPU instructions the i386 CPU does not have.

Any x86 compatible CPU supporting i486, and upward compatible, that is in a *PC compatible* computer should work.

Memory

ttylinux uses an 8 MB RAM disk when booted from CD-ROM, so at least 24 MB of memory are required. The kernel on the CD-ROM is fairly large; it supports a broad range of hardware. Using a custom kernel supporting only hardware for a particular computer, a ttylinux system may require as little as 16 MB of memory. If the file system is installed onto a read/write disk drive, spinning or flash, and a custom kernel is used, ttylinux will run within 8 MB of RAM.

2.1.1 Custom Kernel Requirements

The ttylinux root file system is an 8 MB ext2 file system; the file system image is compressed and resides in the CD-ROM image. After burning the CD-ROM image to a blank CD-ROM disc, or mounting the CD-ROM image via loop device, you can find the compressed root file system; it is `isolinux/filesys.gz`. This root file system can be used with a different custom kernel.

All of ttylinux 8 is built with **Linux 2.6.20** header files. Linux kernels are not backwards-compatible; software using the capabilities of a given kernel version cannot be *expected* to work with any previous kernel version. Using an old kernel cannot be supported in any way. With that described, with the small number of packages in the ttylinux system, ttylinux is known to work to some extent with any Linux kernel from 2.4.4 upwards. The ttylinux firewall script does not work with a 2.2.x kernel.

The kernel you use needs to support all the hardware you want to use, plus some additional requirements for ttylinux itself.

A kernel used for running ttylinux needs to have **ramdisk** support, **initial ramdisk** support, and a default ramdisk size of at least **8192**. Note the kernel configuration has a default ramdisk size of 4096, which is not big enough.

If you want to use the basic firewall script of ttylinux, your kernel also needs iptables support with the **netlink** interface.

If you want to use the telnet server in ttylinux, your kernel will need to have Unix98 pseudo terminal support and support for the **devpts** file system.

A ttylinux kernel needs to support **ext2 file systems**.

2.2 File Downloads

The main ttylinux web site is accessed at <http://ttylinux.org/>. The web site is hosted at <http://www.minimalinux.org/>, which is very much appreciated.

The ttylinux web site has a Download page that has several files available for downloading.

Bootable CD-ROM Image

The bootable CD-ROM ISO image for the i486-PC build of ttylinux is intended to be booted on any PC that can boot from CD-ROM. This file is an image of an El Torito bootable CD-ROM ISO 9660 file system with the Rock Ridge extension. El Torito enables CD-ROM to be bootable on PC. The Rock Ridge extension adds longer file names to the ISO 9660 file system capabilities.

Source Distribution

The ttylinux distribution build system, including its source code packages, is available; it has a file `How_To_Build_ttylinux.txt` that describes the build process.

Binary Run-time Packages

The packages that make up the entire CD-ROM ttylinux run-time are available. Packages are available in the case any were removed from a ttylinux system and there is a desire or need to reinstall the removed packages. After downloading, a package is installed with `pacman`, the ttylinux package manager.

Previous Versions

At least one previous ttylinux version should be available.

2.3 Booting the CD-ROM Image

This version of ttylinux is intended to boot on any x86 PC that can boot from a CD-ROM drive. It can be used as a rescue system or simply for trying ttylinux.

Download the CD-ROM ISO image file and burn it onto a blank CD-ROM disc, as an ISO image. Then boot put the disc into a PC CD-ROM drive and boot the PC; ttylinux should boot up automatically.

A computer's BIOS setup may not be set up to allow booting from CD-ROM; in that case you need to go into the BIOS setup screen(s) and make changes that allow the computer to boot from CD-ROM. If the computer has an old BIOS that is not able to boot from a CD-ROM device, there is software called *Smart Boot Manager* that may help. It can currently be found at: <http://btmgr.sourceforge.net/about.html>

Once ttylinux has booted, and you see the login prompt, login as user name "root", the administrator account, password "password". Read the rest of this user guide for pointers about what

you can do with the system.

After booting, the program `/sbin/ttylinux-installer` can be used to install `ttylinux` from the CD-ROM onto a hard disk or flash disk, such as a USB drive.

2.4 Making a Bootable Flash or USB Disk from the `ttylinux` CD-ROM

Making a bootable flash or USB disk from the `ttylinux` CD-ROM makes a `ttylinux` system different from an *installed* system; when the flash disk is booted, the root file system from the flash disk is decompressed and becomes a read/write root file system in a RAM disk in memory. Changes to files are lost when the system shuts down.

A Linux system, either `ttylinux` or some other Linux system, can be used to make a `ttylinux` bootable flash disk. The `ttylinux` CD-ROM or the CD-ROM image file or even the kernel and file system image files removed from the CD-ROM and in a directory somewhere can be used as the source, depending upon which Linux system is to be used.

The complete process is described in a text file, `Flash_Disk_Howto.txt`, in the `ttylinux` CD-ROM image; you can find it the root (top-level) directory of the CD-ROM.

Also in the root directory of the CD-ROM is a shell script that automates the process of transferring the `ttylinux` system from the CD-ROM onto a flash disk and making the flash disk bootable. The `Flash_Disk_Howto.txt` file has a short description of using this shell script. This shell script depends upon both the CD-ROM and flash disk being mounted; the CD-ROM should be mounted with option `-t iso9660` to specify the correct file system type, and USB disks are usually FAT32 and those should be mounted with option `-t vfat` to specify the correct file system type.

3 Installation Guide

This section describes how to customize and install `ttylinux`.

3.1 Installing from CD-ROM

WARNING: Running the installer can easily destroy all operating systems, and anything else, currently present on the target machine. Proceed with caution and backup all *important* data before installing `ttylinux`. Really.

To install `ttylinux` onto disk from the bootable CD-ROM, you first need to burn the `ttylinux`

CD-ROM ISO image onto a blank CD-ROM disc and boot into it as described in the previous section 2.3.

Once logged in as the "root" user, you can start the installation. You need to know three things to run the installer: what your CD-ROM device is, which disk partition you want to install ttylinux and where you want to put the boot loader.

If you don't know the answers to those three questions after reading the following instructions, the safe bet would be **not** to proceed with installation; the ttylinux installer is not yet automated or user-friendly enough for you.

3.1.1 Source CD-ROM Device

The CD-ROM device depends on whether the drive is an IDE or SATA device. If your system uses IDE, the following device names are possible:

Device Name	Description
/dev/hda	Master Device on First IDE Controller
/dev/hdb	Slave Device on First IDE Controller
/dev/hdc	Master Device on Second IDE Controller
/dev/hdd	Slave Device on Second IDE Controller

Among the above, /dev/hda is not likely to be your CD-ROM device unless you are using a modern laptop. A more likely possibility is /dev/hdc. /dev/hda normally is the device name of your hard disk, but a modern computer will use SATA for the hard drive, but many with IDE CD-ROM drive.

If your system uses SATA (Serial ATA), use this table:

Device Name	Description
/dev/scd0	First SATA CD-ROM Device
/dev/scd1	Second SATA CD-ROM Device
/dev/scd2	Third SATA CD-ROM Device
/dev/scd3	Fourth SATA CD-ROM Device

Usually the SATA CD-ROM device will be /dev/scd0.

3.1.2 Target Partition Device

You need to know, or find out, the device name for the hard disk or flash disk partition on which you want to install ttylinux. The device names for disk partitions are formed by appending

a number to the device name of the corresponding disk. For example, if your disk device is `/dev/hda`, the device `/dev/hda3` is the third partition on that disk. Numbers 1-4 are the primary partitions, extended partitions start at 5.

If you plan on installing onto a USB disk, or some other frequently moved disk device, use the instructions in section 2.4. The disk and partition devices used by this installation process will likely need to be different on a different computer. Because the installation disk and partition device numbers are subsequently used by the installed `ttylinux`, it will not boot, or possibly not work right, when booted on a computer other than the computer on which the installation is performed.

Due to the combined space requirements of the 8 MB `ttylinux` file system and the over 2 MB `ttylinux` kernel, and considering some margin, the minimum partition size you can install `ttylinux` on and have it work is about 12 MB.

IDE disks use the same device names as given for IDE CD-ROM devices above. For SATA, the names are as follows:

Device Name	Description
<code>/dev/sda</code>	First SATA Disk Device
<code>/dev/sdb</code>	Second SATA Disk Device
<code>/dev/sdc</code>	Third SATA Disk Device
<code>/dev/sdd</code>	Fourth SATA Disk Device

Note that if you want to create a dual-boot setup with Windows and `ttylinux` on the same disk, a topic not covered here, you can't use the first partition `/dev/hda1` or `/dev/sda1` as your target partition, because that is where Windows needs to be installed to work.

Here are some examples of possible device names for your target partition:

Device Name	Description
<code>/dev/hda1</code>	First Primary Partition on Primary IDE Master
<code>/dev/hdb5</code>	First Extended Partition on Primary IDE Slave
<code>/dev/sda2</code>	Second Primary Partition on First SATA Disk
<code>/dev/sdc6</code>	Second Extended Partition on Third SATA Disk

Note that depending on the BIOS, booting might be possible only from the first two disks installed in the system.

What to do if your target disk is not partitioned yet? `ttylinux` includes the `fdisk` program that can be used to partition disks. For example, to partition a disk connected as master to the first IDE controller, use:

```
fdisk /dev/hda
```

The user interface of `fdisk` is somewhat primitive, *so be careful*. If you haven't used it before, a good idea would be to search the Internet for instructions. The basic commands you may need are "d" to delete a partition, "n" to create a new partition, "p" to print the current partition table, and "w" to write the edited partition table to disk. You can also use "q" to exit `fdisk` without saving your changes.

3.1.3 Boot Loader Location

The LILO boot loader is installed in one of two places: either the Master Boot Record (MBR) of the *disk device* or the boot sector of the *partition device* in which `ttlinux` is being installed.

With LILO installed in the MBR of the first disk, it will completely take over the entire boot process of the computer. If there are other operating systems installed on the computer they need to be specified in the LILO configuration file, `/etc/lilo.conf`, in order to boot them.

With LILO installed in the boot sector of the target partition or in the MBR of a disk other than the first one in your computer, the bootloader installed in the MBR of the first disk needs to be configured to boot the `ttlinux` target partition.

3.1.4 Running the Installer

Once you have decided on target device and boot loader location, you can run the installer. The script is called `ttlinux-installer` and can be invoked as follows, the square brackets indicate an optional parameter, the *partition device* is used for the installation target:

```
ttlinux-installer [-m] <CD-ROM dev> <target partition dev>
```

For example, to install from the CD-ROM device `/dev/hdc` into partition device `/dev/hda2` and placing LILO on the MBR, `/dev/hda` disk device, you would use:

```
ttlinux-installer -m /dev/hdc /dev/hda2
```

Another example, installing from the second SATA CD-ROM device `/dev/scd1` into the third partition device of the second SATA disk and placing LILO on the boot sector *of the target partition*:

```
ttlinux-installer /dev/scd1 /dev/sdb3
```

The installer checks if the source CR-ROM device contains a `ttlinux` CD-ROM; if the CD-ROM is found then a summary of what is to be installed on which device is printed and you are given

a choice of continuing or aborting. Enter "yes" to continue the installation.

The installer creates an ext2 file system on the target partition then copies the ttylinux distribution files onto the new file system, and then installs the LILO bootloader.

After the installer is finished you can remove the CD-ROM from your computer and reboot.

3.2 Manual Installation

This description uses LILO for boot loading; other boot loaders such as grub and maybe loadlin and syslinux will also work.

There are two files to take from the ttylinux CD-ROM image, either by burning the image to a blank CD-ROM disc and mounting it, or mounting the CR-ROM image via loop device. The ttylinux version 8.0 CD-ROM image file is named `bootcd-i486-8.0.iso.gz`; decompress it and mount it via loop device with the following commands.

```
mkdir -p mnt/ttylinux
gunzip bootcd-i486-8.0.iso.gz
mount -t iso9660 -o loop bootcd-i486-8.0.iso mnt/ttylinux
```

The two files needed from the CD-ROM are the ttylinux root file system, `isolinux/filesys.gz`, and the Linux kernel, `isolinux/vmlinuz`. You can, of course, use a different Linux kernel, following the ttylinux custom kernel requirements described in section 2.1.1.

There are two ways to install ttylinux for booting, one is to have the ttylinux root file system in RAM disk, the other is to install the root file system directory onto a disk.

Install a ttylinux to Boot Using RAM Disk

Copy the ttylinux file system `filesys.gz` image and the desired Linux kernel into your boot files directory; probably, this directory is `/boot`. After copying the two files, unmount the loop device with the following command.

```
umount -d mnt/ttylinux
```

These two files, the kernel and the file system image, can have names other than the file names from the ttylinux CD-ROM. For this example the file names are changed from the names on the CD-ROM: the compressed ttylinux file system image file is called `ttylinux-filesys.gz`, the Linux kernel is called `ttylinux-vmlinuz` and the boot directory is `/boot`. Add the following section to `/etc/lilo.conf`:

```
image = /boot/ttylinux-vmlinuz
label = ttylinux
```

```
initrd = /boot/ttylinux-filesys.gz
root   = /dev/ram0
read-only
```

Run the LILO boot loader installer by typing `/sbin/lilo`. The next boot will have the option of selecting `ttylinux` at the LILO boot prompt.

Install a `ttylinux` to Boot with File System on Disk

A hard disk partition, or a flash disk partition, of at least 8 MB is needed to install `ttylinux`. For this example `ttylinux` is being installed on disk partition device `/dev/hda8` and the kernel and file system files are available via the loop device instructions above. A loop device also is used to mount the `ttylinux` file system image file.

```
cp mnt/ttylinux/isolinux/filesys.gz filesys.gz
umount -d mnt/ttylinux
gunzip filesys.gz
mkdir -p mnt/filesys
mkdir -p mnt/newroot
mount -t ext2 -o loop ./filesys mnt/filesys
mount -t ext2 /dev/hda8 mnt/newroot
cp -aR mnt/filesys/* mnt/newroot
umount -d mnt/filesys
```

The new `ttylinux` root file system is still mounted; it needs to be customized before booting. Customization is described in the following section 3.3; it includes a description of a LILO configuration for booting the new `ttylinux` installation. After customization unmount `mnt/newroot`.

3.3 Customization

This section covers the minimum customization needed to run `ttylinux`. More system configuration and customization can be done; see the system guide, section 5, below for information.

The configuration files and options described in this section are present in a `ttylinux` system installed from the bootable CD-ROM. Following the installation instructions above, the file system is in `mnt/newroot`, that is the example starting point for the following customization descriptions.

/etc/fstab

`/etc/fstab` needs to have the correct device for the root directory, the manually installed `ttylinux` `/etc/fstab` still specifies a RAM disk device for the root directory. Change the RAM disk device, `/dev/ram0`, to be the disk partition device in which the `ttylinux` root file system was installed. In the above example `/dev/hda8` was used, so for that example the root directory in `/etc/fstab` would be specified as:

```
/dev/hda8      /          ext2      defaults    0 0
```

Boot Loader

The boot loader needs to be aware of the ttylinux kernel and root file system device. Following the installation instructions above, the LILO configuration file `/etc/lilo.conf` would include the following. Note the `initrd` specifier is removed and the `root` specifier is changed to `/dev/hda8`.

```
image = /boot/ttylinux-vmlinuz
       label = ttylinux
       root = /dev/hda8
       read-only
```

Keyboard Map and Timezone

To use the current keyboard map and timezone from the Linux computer being used to install ttylinux, use the following commands.

```
rm mnt/newroot/etc/localtime
cat /etc/localtime > mnt/newroot/etc/localtime
mnt/newroot/bin/dumpkmap > mnt/newroot/etc/i18n/kmap
```

Dial-up Network Information

[The ttylinux PPP and ISDN structure is being re-organized.]

To set up the dial information for an Internet provider, you need to know the following:

- the device your modem is connected to, i.e. `/dev/ttyS1`
- the init string for your modem, i.e. `ATZ`
- the speed your modem can talk to the computer, i.e. `115200`
- the phone number of your ISP, i.e. `1-800-LETMEIN`
- your user name at the ISP, i.e. `joeuser`
- your password at the ISP, i.e. `joepass`

The above is for dialup using a modem. If you want to use ISDN dialup, you need to know the following:

- the phone number of your ISP, i.e. `1-800-12345678`
- your user name at the ISP, i.e. `joeuser`

- your password at the ISP, i.e. joepass
- the phone number you want to use for dialing out
- whether you want to use automatic or manual dialing

Edit the file `mnt/newroot/etc/ttyisp.conf` and make the appropriate changes. `tylinux` works only with providers that launch a PPP session immediately after dialup, which should be true for almost all providers; if a different procedure is needed to login, your Internet connection will not work with `tylinux`. The `tylinux` approach will work with almost all providers in Germany. [The creator and original maintainer of `tylinux` is Pascal Schmidt who lives in Germany.]

For `tylinux` to use ISDN, you need to have `USE_ISDN=yes` in `/etc/ttyisp.conf`, the default is not to use ISDN. Also needed is a kernel with ISDN support and hardware support for your ISDN hardware. If you need kernel modules to run your ISDN hardware, you will also need to add those and a suitable `/etc/modules.conf` to the `tylinux` file system.

4 Add-ons

Add-ons may be available at the `tylinux` web site.

Previous `tylinux` add-ons may no longer be supported and no longer be available at the `tylinux` web site. Previous add-ons may become unsupported because they do not properly build in the `tylinux 8` build environment, or because their run-time environment is not supported in the `tylinux 8` run-time environment.

Corrected and new add-ons submitted to `tylinux` will be considered for inclusion at the web site.

5 System Guide

This section gives a short overview of the `tylinux` system, its configuration and some of the installed programs.

5.1 Basic Features

Upon boot-up, `tylinux` provides 6 text consoles for login. There is one initial user account: `root`, the administrator account, with password `password`.

The `syslogd` and `klog` daemons are running and logging kernel and system messages to the file `/var/log/messages`.

The available text editor is `vi`; invoke it by typing `vi /path/to/filename`. This version of `vi` is a minimal version provided by `busybox`. Documentation and help for using `vi` is available in many places on the web.

For manipulation of users, groups and passwords, the tools `adduser`, `addgroup`, `deluser`, `delgroup` and `passwd` are present.

If you have not changed the timezone and keyboard settings as outlined in the customization section, section 3.3 above, `ttlinux` will use its default settings. The default timezone is UTC, the default keyboard mapping is for a US keyboard.

The `inetd` super-server and the `dropbear` SSH server are running by default. A telnet server will be forked by `inetd` when a telnet connection comes in.

`ttlinux` includes a basic packet filtering firewall which is enabled by default on the dialup interface. Note that the telnet server is not visible from the external side of the dialup interface in the default firewall configuration.

5.2 Bootup, Shutdown and System Configuration

On system bootup, the `init` process runs the `/etc/rc.d/rc.sysinit` script to setup the system, such as setting the clock, system font, keyboard map and checking the file systems. `rc.sysinit` also runs all the programs in the `/etc/rc.d/rc.startup` directory *and then* runs the `/etc/rc.d/rc.local` script, all with the command line parameter `start`.

On system shutdown, the script `/etc/rc.d/rc.sysdone` runs. This script runs all the programs in the `/etc/rc.d/rc.shutdown` directory *and then* runs the `/etc/rc.d/rc.local` script, all with the command line parameter `stop`.

All the programs in `/etc/rc.d/rc.startup` and `/etc/rc.d/rc.shutdown` are symbolic links that reference actual shell scripts or binary programs; they are run in the ASCII order of their file names. These symbolic links are named with leading numbers to help control their ordering e.g., `10.network` is the symbolic link to the network startup program. The actual programs are in `/etc/rc.d/init.d`. Removing a symbolic link disables the program from starting up. These programs typically are shell scripts; they are commonly called *initscripts*.

Initscripts can be interactively invoked. The following command runs the network script `/etc/rc.d/init.d/syslog` with the command line option `stop`.

```
service syslog stop
```

All scripts use the command line options `start`, `stop`, `reload`, `restart` and `status`. They print a list of supported options if they are called with no option present.

The initscripts define the basic `ttlinux` bootup system configuration. The initscripts are configurable, to an extent; thus the bootup configuration is configurable, to an extent. The bootup system configuration is specified in ASCII text files in the `/etc/sysconfig` directory; this directory is intended to have only files that are read by the various initscripts. All files read by initscripts for configuration options should reside in `/etc/sysconfig`.

5.3 Shell Environment

The default shell used by `ttlinux` is GNU `bash`. The shell environment of aliases and variables is in `/etc/bashrc`; view this file after login to become familiar with the default shell environment.

Upon login, the `PATH` environmental variable has the following paths in the order listed.

```
/bin
/usr/bin
/sbin
/usr/sbin
```

Put additional, or overriding, shell environment in scripts in the `/etc/profile.d` directory; do not change `/etc/bashrc` in order to avoid losing changes when updating `ttlinux`.

5.4 Using the Dropbear for SSH

SSH, or secure shell, is a protocol for remote login with an advantage over `telnet` being that it can use public key authentication instead of passwords. Another advantage over the `telnet` protocol is that plain text is not transferred; the data sent between the host connections is encrypted.

`dropbear` is a small SSH v2 server and client package. The server is started on system bootup by default. It allows password and public key authentication. Public key authentication can use DSS and RSA keys and works with keys generated by the popular `OpenSSH` package. Having a public key from `OpenSSH` in the file `.ssh/authorized_keys` should allow secure login from the machine that has the corresponding private key. The permissions on the `.ssh` directory must not include group or other write permission, otherwise `dropbear` will refuse public key authentication.

The SSH client program is called `dbclient`. It is different from the server in that it cannot use keys in `OpenSSH` format. You can use the `dropbearconvert` program to convert an `OpenSSH` format key for use by `dbclient` or you can use `dropbearkey` to create a new key.

To convert an `OpenSSH` key stored in `~/.ssh/id_rsa`, do:

```
dropbearconvert openssh dropbear \  
  ~/.ssh/id_rsa ~/.ssh/id_rsa.db
```

The new key will be stored in `~/.ssh/id_rsa.db`. You can use the `-i` switch to `dbclient` to make it use your new key for authentication. The public key part of the old OpenSSH key can be used as-is for pasting into your `~/.ssh/authorized_keys` file. Conversion is only needed for the private key.

To create a new RSA key to store in `~/.ssh/id_rsa.db`, you can use the following command:

```
dropbearkey -t rsa -f ~/.ssh/id_rsa.db
```

The public key part of the new key will be printed to the screen. You can put it into the `~/.ssh/authorized_keys` file on all machines where you want to be able to login using your new private key stored in `~/.ssh/id_rsa.db`. You can create a DSS key instead of an RSA key by using `-t dss` instead of `-t rsa`. Should you lose the public key, you can always get it back by using the private key and the `-y` switch to `dropbearkey`:

```
dropbearkey -y -f ~/.ssh/id_rsa.db
```

If you want to use `scp` to copy files from another machine, the standard `scp` program from OpenSSH is included with `dropbear` and `ttylinux`.

5.5 Using an Ethernet Network

`ttylinux` is ready to use Ethernet networking. DHCP will be used when starting up the Ethernet network, unless configured otherwise.

The Ethernet network interface configuration is specified in the text file:

```
/etc/sysconfig/network-scripts/ifcfg-eth0
```

This file has specification in the form of "ITEM=value". Edit this file to set the proper Ethernet interface IP addresses, change the Ethernet DHCP usage and to enable Ethernet networking. To enable Ethernet networking, the line `ENABLE=no` must be changed to `ENABLE=yes`. To disable DHCP, the line `DHCP=yes` must be changed to `DHCP=no`.

After configuring the Ethernet network interface, restart the networking subsystem with the following command.

```
service network restart
```

See the description of the `/sbin/sysconfig` script in section 5.9 for scripted help in setting up the Ethernet network interface configuration.

The Ethernet network interface, commonly referred to as `eth0`, can be started and stopped independently from the entire network subsystem with the following commands.

Startup `eth0` with:

```
ifup eth0
```

Shutdown `eth0` with:

```
ifdown eth0
```

5.6 Using the Firewall

[The `ttylinux` firewall structure is being re-organized.]

The default settings of the `ttylinux` firewall script assume that the network interface to the rest of the world is called `ipp0` or `ppp0`, depending on whether you use ISDN or modem dialup. By default, the firewall will allow the world to access the SSH server, but will deny access to the telnet server. Other incoming connection attempts will also be denied and logged, rate limited to an average of 3 messages per hour, except for connections to the `ftp-data` port, which is used for active FTP sessions and allowed in. This applies to TCP connections. For the UDP protocol, incoming packets to ports above 1024 are allowed. Also, the outside world is not allowed to ping the `ttylinux` machine.

Outgoing traffic is not firewalled at all.

You can change the settings of the firewall in `/etc/firewall.conf`. The only configuration options are whether to allow or deny access to the SSH and telnet servers.

Note that this firewall script is very simple and assumes that local users are always trusted. There is no protection against users running server daemons on UDP ports above 1024. If you don't trust your users, this firewall script is not good enough. Also note that only the `ppp0` or `ipp0` device is protected, all other interfaces are not firewalled at all.

5.7 Using Dialup

[The `ttylinux` PPP and ISDN structure is being re-organized.]

What to put into `/etc/isp.conf` to configure the dialup information is outlined above in the customization section (3.3, above). What you will find here is an overview of how to actually use the PPP subsystem for Internet dialup.

Assuming `/etc/isp.conf` was setup correctly before boot, you only need one command to start up the Internet connection:

```
ppp-up
```

If you use ISDN with autodialing enabled (`ISDN_MODE=auto` set in `/etc/isp.conf`), you do not need to run `ppp-up`. Just run a program that sends packets to an Internet host.

`ppp-up` triggers the PPP daemon, which will dial the provider and try to login. `ppp-up` can be run by any user in the `ppp` group. In the default `ttylinux` configuration, the `ppp` group has no members, so only `root` can start and stop the internet connection. You can watch `/var/log/messages` (using `tail -f`) to see when the connection is up and running. After that, the Internet connection is ready for use. For example, the SSH server will be reachable from the outside world.

You can browse the web by using the `retawq` web browser. Use the `g` key to enter a URL, then navigate around using the cursor and return keys. You can leave the browser by pressing `Shift-Q`.

If you use ISDN, you can enable and disable channel bundling by calling the `bundle` and `unbundle` scripts while the connection is up. The default for new connections is to always use a single channel, even if channel bundling was enabled before. To check whether channel bundling is currently enabled, use the command:

```
service isdn status
```

If you decide that you want to cut the Internet link, just issue:

```
ppp-down
```

This will take the connection down. Once again, this can be done by `root` or a user from the `ppp` group, in this case only the user who started the connection.

If you change any of the settings in `/etc/isp.conf`, you need to update the PPP configuration files by running the following command:

```
ppprestart
```

Note that if you want to switch from using a modem to using ISDN or vice versa, you need to take down the Internet connection before running `ppprestart`.

After that, you can again use `ppp-up` and `ppp-down`. Note that running `ppprestart` only works for `root`, not ordinary users. In previous `ttylinux` versions (2.5 and older), rebooting also updated the PPP configuration files. This is no longer the case to allow manual modifications of the files. They will only be created on boot if they don't exist.

Users in the `ppp` group will be able to read the file `/etc/isp.conf` which contains your dialup password. This may or may not be a good idea, depending on how much you trust your users. If you only wish to use ISDN, you can remove the group read permission with

```
chmod g-r /etc/isp.conf
```

5.8 Package Management

Package management is handled by shell script `/usr/bin/pacman`; since the directory `/usr/bin` is in the `PATH`, `pacman` is invoked from the shell command line by typing its name, `pacman`. Its options are somewhat similar to Red Hat's package manager `RPM`. Use `pacman` to install and remove packages, and to query the database of installed packages and files.

The binary packages used by `ttylinux` are `tar` archives compressed with the `bzip2` algorithm. All the packages that normally come with the `ttylinux` distribution are available on the download page at the `ttylinux` web site; this is for reinstalling any packages that may have been removed from a `ttylinux` system.

To install the package `bash-3.2-i486.tbz`, you would use the following command:

```
pacman -i pkg-bash-3.2.tbz
```

`pacman` can install a package from standard input. This is useful for forming a pipe with the `wget` program, which allows installing a package from the network without having to store it. For example, use the following to download and install the `bash` package from a `ttylinux` web site `ttylinux.net`:

```
wget http://ttylinux.net/Download/bash-3.2-i486.tbz -O - | pacman -i -
```

Three different options are available for querying the database of installed packages and files. To get a list of all installed packages, run

```
pacman -qa
```

To find out which package the file `/bin/login` belongs to, use

```
pacman -qf /bin/login
```

To list all files from the `e2fsprogs-1.41.3` package, run

```
pacman -ql e2fsprogs-1.41.3
```

If you want to remove a package, you can do so by use of the `-e` option. To remove the `isd4k-utils.v3.2p1` package, you would use the following command:

```
pacman -e isdn4k-utils.v3.2p1
```

You can also use the `-v` option to get verbose output during installation and removal of packages. `pacman` will then list all the files it has installed or removed.

pacman Database

The `pacman` script uses directory `/usr/share/ttylinux` which has one file per installed package, each file containing a listing of all the files that belong to the installed package.

5.9 Using the `sysconfig` Script

The `/sbin/sysconfig` shell script can be used to set, and to show, the fields in various `ttylinux` system configuration files; it can set or show any value for any "ITEM=value" line in any configuration file in the `/etc/sysconfig` and `/etc/sysconfig/network-scripts` directories.

The following commands sets "ENABLE=yes" and "DHCP=yes" in the `/etc/sysconfig/network-scripts/ifcfg-eth0` file.

```
sysconfig -nc ifcfg-eth0.enable=yes
sysconfig -nc ifcfg-eth0.dhcp=yes
```

The "-nc" option in the above examples tells the `sysconfig` script to work on files in the `/etc/sysconfig/network-scripts` directory. The second option is in the form `file.item=value`.

To change the IP address of the Ethernet network interface, with 192.168.1.100 as the example IP address, with a netmask of 255.255.255.0 and standard subnet gateway and broadcast addresses, use the following sequence of `sysconfig` script commands.

```
sysconfig -nc ifcfg-eth0.ipaddress=192.168.1.100
sysconfig -nc ifcfg-eth0.network=192.168.1.0
sysconfig -nc ifcfg-eth0.netmasks=255.255.255.0
sysconfig -nc ifcfg-eth0.gateway=192.168.1.1
sysconfig -nc ifcfg-eth0.broadcast=192.168.1.255
```

Use "-sc" for the first option to the `sysconfig` script in order to work with system configuration files in the `/etc/sysconfig` directory.

Use the following command to get complete, up-to-date help description directly from `/sbin/sysconfig`

```
sysconfig --help
```

5.10 Deprecated and Legacy Items

5.10.1 devfs

Support for devfs was removed from ttylinux Version 4.0, in Aug 2004. There is no current plan to use a dynamic device file system.

6 Contact and Help

Reporting bugs in ttylinux and its documents is appreciated. For bug reports, suggestions, or anything else about ttylinux that you think is important, feel free to contact me. You can reach me by email at:

Douglas Jerome <douglas@ttylinux.org>

There is a web-based forum that is active from time to time; it is active when this was written, December 2008, and is intended to be active as long as minimalinux is supporting ttylinux, barring spammer abuse.

<http://www.minimalinux.org/forum/>

Help may be available on irc, although it is very low bandwidth and usually more appropriate for inane banter.

`irc.freenode.net #ttylinux`

A ttylinux-specific Commands Overview

Separate from the initscripts in `/etc/rc.d/initd` directory, the following table lists the ttylinux-specific scripts intended to be available for ttylinux root users.

Script	Directory	Usage
ttylinux-flash	CD-ROM	Copy ttylinux to Flash Disk
ifdown	/sbin	Shutdown Ethernet Network Interface
ifup	/sbin	Startup Ethernet Network Interface
service	/sbin	Execute a script in <code>/etc/rc.d/init.d</code>
sysconfig	/sbin	Modify a System Configuration File
ttylinux-installer	/sbin	Install ttylinux onto A Disk
pacman	/usr/bin	ttylinux Package Manager