

ttylinux User Guide

Version 7.0

Pascal Schmidt

March 9, 2008

Contents

1	Introduction	4
1.1	What is ttylinux?	4
1.2	What is ttylinux not?	4
1.3	Who is ttylinux aimed at?	5
1.4	License	5
2	Installation guide	6
2.1	Requirements	6
2.1.1	Custom kernel requirements	6
2.2	Download	7
2.3	Booting the CD-ROM version	8
2.4	Installing from CD-ROM	8
2.4.1	Source CD-ROM device	8
2.4.2	Target partition device	9
2.4.3	Boot loader location	10
2.4.4	Running the installer	11
2.5	Customization	12
2.6	Installation by hand	14
3	System guide	15
3.1	Basic Features	15
3.2	Bootup and Shutdown	16

Contents	3
3.3 Using PPP dialup	17
3.4 Using the dropbear SSH package	18
3.5 Setting up network devices	19
3.6 Using the firewall	20
3.7 Shell environment	20
3.8 Package Management	21
3.9 Using devfs (no longer supported)	22
4 Contact	22
A How to create a bootable ttylinux CD	22

1 Introduction

This section gives an overview on ttylinux. What is it, who can use it, why it is there.

1.1 What is ttylinux?

ttylinux is a minimalistic Linux distribution. It can be installed by hand or by using the bootable CD version that includes an installer script. For a custom installation by hand, another Linux system already up and running is needed.

ttylinux aims to put as much as possible into as little space as possible. It provides multi-tasking, multi-user, and networking capabilities within less than 8 megabytes of operating system size.

The main use that ttylinux comes prepared for is dialing into an Internet provider by modem or ISDN. A text mode web browser is present.

A package management script capable of installing and removing binary software packages is included. This makes it possible to adapt ttylinux to specific needs.

ttylinux can be installed onto hard disk, CD-ROM, or any other media you can boot from. With a custom made bootable CD-ROM, as little as 4 MB of space are enough. The basic filesystem (excluding the kernel) is 8 MB in size. If you use the installer provided by the bootable CD-ROM version of ttylinux, a kernel image of a little under 2 MB will also be installed. This makes the minimum workable size for a hard disk partition to install ttylinux on around 10-12 MB.

You can also use ttylinux as a basis for putting together a more complete minimal system. ttylinux uses the same C library as full Linux systems, so compiling programs on another Linux machine and copying them over to the ttylinux filesystem should result in working programs.

1.2 What is ttylinux not?

ttylinux is not as complete a Linux distribution as others, mostly due to the limited size. It does not contain developer tools or a graphical user interface (GUI), for example.

1.3 Who is ttylinux aimed at?

As ttylinux can be used on machines which are considered obsolete by now, such as 486SX class PCs, it is aimed at people who want to have a minimalistic Linux distribution to run when little space is available.

ttylinux is aimed at people who already have some Linux experience. There is no big user manual describing how to use the system, so a bit of Unix or Linux experience is very useful when trying to use the system. Most of the programs present are smaller versions of common Unix utilities, and documentation on those can be found in various places on the Internet.

ttylinux can fit into an Eltorito image useful for bootable CD-ROMs. It could be used for installation discs or rescue systems on CD. ttylinux provides a working Linux environment from the boot image, and then a script could mount the rest of the CD to provide a full system.

Note: ttylinux is really not aimed at Linux beginners. Knowledge about how to compile a kernel (the basic ttylinux filesystem comes without a precompiled kernel to allow maximum flexibility, though the CD-ROM version provides a precompiled kernel) and find your way around a Linux system in general is assumed.

1.4 License

The software packages that are part of ttylinux are licensed under a number of different open source licenses, as listed below. The scripts that hold the distribution together are licensed under the GNU General Public License. A copy is included in the file COPYING.

Package	Version	License
bash	3.2.33	GPL
busybox	1.9.1	GPL
dropbear	0.50	MIT
e2fsprogs	1.40.7	GPL
glibc	2.7	LGPL
iptables	1.4.0	GPL
isdn4k-utils	3.2p1	GPL
lilo	22.8	BSD
ncurses	5.6	MIT
ppp	2.4.4	BSD
retawq	0.2.5c	GPL

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

2 Installation guide

This section describes the requirements of ttylinux, where to download it, and how to customize and install it.

2.1 Requirements

The ttylinux distribution requires at least a 486SX processor. How much memory is needed depends on how you want to run the system. Running the system from ramdisk will need about 14-16 MB of memory, depending on the exact kernel version used.

The bootable CD-ROM version uses a fairly large kernel image to support a broad range of hardware. It also loads the whole system into a ramdisk. This means the bootable CD-ROM version needs about 16-18 MB of memory to run.

Once the system is installed onto hard disk, the memory requirement is reduced to about 8-10 MB of memory. This is when using the default kernel as installed by the CD-ROM installer. This requirement can be reduced further by installing a custom kernel by hand. With a minimal 2.4.x kernel configuration, for example, ttylinux can be made to run within 5-6 MB of memory.

2.1.1 Custom kernel requirements

If you install ttylinux by hand, not using the bootable CD-ROM, what you get is a 8 MB ext2 filesystem image that contains only the distribution files. You will need to add a Linux kernel to this to make a complete system.

The kernel you use will obviously need to support all the hardware you want to use, plus some additional requirements for ttylinux itself. The kernel used for running ttylinux needs to have ramdisk support, initial ramdisk support, and a default ramdisk size of at least 8192. Note that the stock kernel configuration will use a ramdisk size of 4096, which is not enough. If you want to use the basic firewall script of ttylinux, your kernel also needs iptables support.

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` filesystem.

`ttylinux` is basically designed to work with any kernel from 2.4.4 upwards. You might be able to run the system with a 2.2.x kernel, but this is neither officially supported nor tested. The `ttylinux` firewall script definitely won't work with a 2.2.x kernel.

The Linux kernel on the system that is used to install `ttylinux` needs to have support for the `ext2` filesystem and loopback device. If there is no loopback support, you cannot customize `ttylinux` for your system. No problem if the defaults suit you perfectly, but that is rather unlikely.

For the default installation process describes below, it is also required that the system used to install `ttylinux` is based on `glibc`, also known as `libc6`. The customization process requires you to run a program from `ttylinux` at one point that is linked against `glibc`.

This may all sound like a lot of requirements for the preparation system, but almost any half-recent Linux distribution with its default kernel will do.

2.2 Download

You can download `ttylinux` from the official homepage, which at the moment is at <http://www.minimalinux.org/ttylinux/>.

Go to the downloads page and get the ISO image version if you want to burn `ttylinux` onto a bootable CD-ROM. This version of `ttylinux` can be booted on any PC that can boot from a CD-ROM drive. You can use it as a rescue system or for trying out `ttylinux`. You can also use the installer script to install the system onto hard disk after booting from the CD-ROM.

If you want to install by hand, you can find a binary tarball of the distribution on the downloads page. Grab this. Inside the archive you will find this user guide and a file called `rootfs.gz`, which is the compressed filesystem image for `ttylinux` itself. It contains an `ext2` filesystem with the distribution installed on it.

The individual binary packages that were used to create the filesystem image are also available for download. You may want to download them if you use the package manager to remove a package from your `ttylinux` system and later need to reinstall it.

You can also download source code packages, including a build system that you can use to build a binary `ttylinux` package yourself. The build system package includes a `README` file that describes the process.

2.3 Booting the CD-ROM version

To boot ttylinux from CD-ROM, simply burn the downloaded ISO image onto a blank disc. Then reboot your computer with the new disc inserted into your CD-ROM drive. ttylinux should boot up automatically. Note that your computer's BIOS setup may not be set up to allow booting from CD-ROM discs. In that case you need to go into the BIOS setup screen(s) and change that.

If your computer has an old BIOS that does not know how to boot from a CD-ROM device, there is a software called "Smart Boot Manager" that may work for you. It can currently be found at:

```
http://btmgr.sourceforge.net/about.html
```

Once ttylinux has booted up, you will see a login prompt. You can use the user name "root" (the administrator account) and password "root" for logging in. Read the rest of this user guide for pointers about what you can do with the system.

2.4 Installing from CD-ROM

WARNING: Running the installer can easily destroy all operating system installations currently present on the target machine. Proceed with caution and backup all important data before installing ttylinux! This is no drill.

To install ttylinux onto hard disk from the bootable CD-ROM, you first need to burn the ISO image onto a blank CD-ROM disc and boot into it as described in the previous section.

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, onto which hard 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 – sorry, the ttylinux installer isn't yet automated or user-friendly enough for you.

2.4.1 Source CD-ROM device

The name of your CD-ROM device depends on how the drive is hooked up to your IDE or SATA controller. 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 very unlikely to be your CD-ROM device (this is normally the device name of your hard disk). A more likely possibility is /dev/hdc.

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

Most of the time, your SATA CD-ROM device will be /dev/scd0.

2.4.2 Target partition device

You need to find out the device name for the hard disk partition you want to install ttylinux on. The device names for partitions are formed by appending numbers to the device name of the corresponding hard disk. For example, if your hard disk device is /dev/hda, the device /dev/hda3 is the third partition on that hard disk. Numbers 1-4 are the primary partition, extended partitions start at 5.

Due to the combined space requirements of the 8 MB ttylinux filesystem and the 2 MB default kernel, the minimum partition size you can install ttylinux on and have it work is about 10-12 MB.

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

Device name	Description
/dev/sda	First SATA hard disk device
/dev/sdb	Second SATA hard disk device
/dev/sdc	Third SATA hard disk device
/dev/sdd	Fourth SATA hard disk device

Note that if you want to create a dual-boot setup with Windows and `ttylinux` on the same hard-disk later (a topic not covered here), you can't use `/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 hard disk
<code>/dev/sdc6</code>	Second extended partition on third SATA hard disk

Note that depending on the BIOS, it might not be possible to boot from anything but the first two hard 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 hard disks. For example, to partition a hard 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.

2.4.3 Boot loader location

You can install the LILO boot loader in one of two places: either into the master boot record (MBR) of the hard disk you are installing to or into the boot sector of the partition you are installing to.

With LILO installed in the MBR of your first hard disk, it will take over the boot process of your computer completely. If you have other operating systems installed on the computer, you will need to add them to the LILO configuration file (`/etc/lilo.conf`) in `ttylinux` to be able to boot into them.

With LILO installed in the boot sector of the target partition or in the MBR of a hard disk other than the first one in your computer, you will need to have another boot loader installed in the MBR of your first hard disk. You have to add the MBR or boot sector where the ttylinux LILO is installed to the setup of that boot loader in order to be able to boot ttylinux.

2.4.4 Running the installer

Once you have decided on target device and boot loader location, you can run the installer. The script is called `installer` and can be invoked as follows (the square brackets indicate an optional parameter):

```
installer [mbr] source-device target-device
```

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

```
installer mbr /dev/hdc /dev/hda2
```

Another example, installing from the second SATA CD-ROM device `/dev/scd1` onto the third partition of the second SATA hard disk, placing LILO in the boot sector of the target partition:

```
installer /dev/scd1 /dev/sdb3
```

The installer will check if your source device contains a ttylinux CD-ROM before proceeding. If the CD-ROM is found, a summary of what is going to be installed where is printed and you are given a last chance of aborting the whole install. Enter "yes" to continue the installation or anything else to abort.

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

Once the installer is finished, it prints out a success message. At this point you can remove the CD-ROM from your computer and reboot.

2.5 Customization

The following instructions are mainly useful for installing ttylinux by hand, but the configuration files and options described here are also present in a ttylinux system installed from the bootable CD-ROM.

Before you go about installing ttylinux, you may want to change a few settings inside the image file to fit your own system. This is done by uncompressing and mounting the image and then making the appropriate changes. If the image is in the current working directory, do the following as the root user:

```
gunzip rootfs.gz
mkdir mnt
mount -o loop rootfs mnt
```

Now the ttylinux filesystem is mounted on the directory `mnt`. Take a look at it if you wish. If you want to install ttylinux to a hard disk by hand, you can now create an installation archive for that purpose as follows:

```
cd mnt
tar cvzf ../ttylinux-hd.tar.gz .
cd ..
```

After this, you have a file called `ttylinux-hd.tar.gz` that can be unpacked onto any filesystem you like.

The things you will probably want to customize on ttylinux are the keyboard map and timezone in use and the Internet provider settings. You can copy over the keyboard and timezone setting from your working system as follows:

```
mnt/bin/dumpkmap > mnt/etc/i18n/keymap.dat
cat /etc/localtime > mnt/etc/localtime
```

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/etc/isp.conf` and change the settings there to your liking. Note that `ttylinux` only knows about providers that launch a PPP session immediately after dialup (which should be true for almost all providers these days), so if some special procedure is needed to log you in, your Internet connection will not work with `ttylinux`. The `ttylinux` approach will work with almost all providers in Germany, but I don't know about other countries.

For `ttylinux` to use ISDN, you need to have `USE_ISDN=yes` in `/etc/isp.conf`, the default is not to use ISDN. Obviously, you also need 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 `ttylinux` filesystem.

Once you are finished making changes, unmount the filesystem and run `gzip` to compress it again:

```
umount mnt
gzip -9 rootfs
```

You are now ready for the real installation.

Note that this section has only covered the minimum customization needed to run `ttylinux`. There is more to tune, have a look at the system guide (section 3) below for information. If you aim at the approach of running `ttylinux` from ramdisk, you must configure everything for your needs before you continue the installation, so it may be a wise idea to read the system guide section before you continue. If you install to a hard disk, you can always configure things later.

2.6 Installation by hand

I am assuming that you use LILO (short for Linux Loader) as your boot loader. You can also use others like `loadlin`, but you will obviously need some LILO experience to convert the LILO example given here to some other boot manager. I am also assuming that you want to add `ttylinux` to an existing LILO setup that is already used to boot other (Linux) systems on your machine.

If you want to use the `syslinux` boot loader to load `ttylinux` from a FAT filesystem, please see the appendix on creating a bootable CD-ROM (appendix A) for an example.

Put the `rootfs.gz` image and the Linux kernel you want to use to run `ttylinux` into the directory that holds your other boot files, most likely this will be `/boot`. Let us assume your kernel image file is called `bzImage-ttylinux`. You now need to add the following section to `/etc/lilo.conf`:

```
image=/boot/bzImage-ttylinux
label=ttylinux
initrd=/boot/rootfs.gz
root=/dev/ram0
read-only
```

Now rerun the LILO installer by typing `/sbin/lilo`. On your next boot, you can enter `ttylinux` on the LILO boot prompt to boot `ttylinux`. That is all, you are now ready to try `ttylinux`.

If you want to install onto hard disk, you need to have a partition of at least 8 megabytes size available. Assuming this partition is `/dev/hda8`, do the following, using the file `ttylinux-hd.tar.gz` obtained above (in section 2.5):

```
mke2fs /dev/hda8
mount -t ext2 /dev/hda8 mnt
cd mnt
tar xvzf ../ttylinux-hd.tar.gz
cd ..
```

Now do the customization as outlined above. You will also need to change the root device configured in `mnt/etc/fstab`. Once you are done, you just need to unmount the partition:

```
umount mnt
```

Now you can setup LILO as explained above, only change the line for the root filesystem to

`root=/dev/hda8` and remove the `initrd` line. Ready to go.

3 System guide

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

3.1 Basic Features

On bootup, `ttylinux` provides 6 text consoles for login. The default setup has two user accounts: `root` (the administrator account) with password `root` and `guest` with password `guest`.

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

If you need a text editor, `vi` is installed. Simply invoke it by typing `vi /path/to/filename`. This version of `vi` is a minimalist version provided by `busybox`. You can find documentation and help about using `vi` in many places on the web.

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

If you have not changed the timezone and keyboard settings as outlined in the customization section (2.5, above), `ttylinux` 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.

`ttylinux` 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.

3.2 Bootup and Shutdown

On system bootup, the `init` process runs the script `/etc/rc.d/rc.sysinit` to get the system up. The script will do the usual business of checking and remounting the root filesystem and will start up basic networking.

After that is done, `rc.sysinit` will run all executables from the directory `/etc/rc.d/rc.start`. It does not matter whether these are scripts or compiled programs. All executables will be passed the parameter `start`. For example, if there is a script called `runme`, `rc.sysinit` will execute `runme start`.

The default initscripts that come with `tylinux` are stored in the directory `/etc/rc.d/init.d`. Those that need to be run at bootup are symlinked into the `rc.start` directory. If you want to write your own initscripts, place them in the `init.d` directory and place a symlink in the `rc.start` and perhaps also in the `rc.stop` directory.

If you don't want to write your own script, you can also add commands to the file `/etc/rc.d/rc.local` - this script will be run at the very end of the boot process, after all others scripts have run.

On shutdown, the script `/etc/rc.d/rc.reboot` gets run. This will run all the programs from the `/etc/rc.d/rc.stop` directory, and each program will be passed a `stop` parameter. Our example script called `runme` would be executed as `runme stop`.

Both scripts from `rc.start` and `rc.stop` will be run in ASCII order. If you need them to run in a specific order, you could for example use names like `01.first` and `02.second`. This is how the scripts that come with `tylinux` are named. Everything in `rc.start` and `rc.stop` should be a symlink, so you can simply remove that link if you want to disable a specific script.

Initscripts can also be called manually in case you want to start or stop something while the system is running. To make this easier, a helper script called `service` is provided. For example:

```
service inetd stop
```

will stop the `inetd` server daemon. All scripts (except `pppconf`, which understands few options) know about the options `start`, `stop`, and `restart` that do the obvious things. Some scripts also know about `status` and `reload` for displaying status information and for reloading service configuration files.

All scripts will print a list of supported options if they are called with no option present.

3.3 Using PPP dialup

What to put into `/etc/isp.conf` to configure the dialup information is outlined above in the customization section (2.5, 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
```

3.4 Using the dropbear SSH package

SSH, or secure shell, is a protocol that allows remote logins to a machine. It uses encrypted connections and can use public key authentication instead of plain passwords. The advantage over the older telnet protocol is that nothing in an SSH session goes over the network unencrypted.

`dropbear` is a small SSH v2 server and client package. The server will be 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. Putting your public key from OpenSSH into the file `.ssh/authorized_keys` should be enough to let you login from the machine where you keep the corresponding private key. Please note that 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`.

3.5 Setting up network devices

Basic network settings are located in the `/etc/network.conf` file. In that file, you can globally disable or enable networking. Also, it is possible to specify a gateway address. If configured, `ttylinux` will setup a default route to the gateway on bootup. This is mainly useful if you don't use modem or ISDN dialup. If you do use dialup, the default route will be overwritten once you start a dialup connection.

Note that loopback networking via the `lo` device is always enabled. The setting in `/etc/network.conf` only covers devices that can contact the outside world.

To set up a network device, go to the directory `/etc/network.d` and copy the file `sample` found there to a file named `interface.dev`. For example, if you want to setup the device `eth0`, name the file `interface.eth0`:

```
cp sample interface.eth0
```

Now edit the new file to adjust the settings for the device. You can either let the system use the

DHCP protocol to obtain the settings from a DHCP server or you can set the IP address and other parameters manually within the file.

3.6 Using the firewall

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 enough for you! Also note that only the `ppp0` or `ipp0` device is protected, all other interfaces are not firewalled at all.

3.7 Shell environment

The default shell used by `ttylinux` is GNU `bash`. You can change the startup environment (variables, aliases, etc) in `/etc/profile`. You can also put files into the directory `/etc/profile.d` – all readable files in that directory will be sourced by `bash` when you login on the `ttylinux` machine.

Using `/etc/profile.d` is a good choice for your own personal changes. You may want to update to a new version of `ttylinux` sometime later, and you could end up losing all your changes to `/etc/profile` if you forget to make a backup. In `/etc/profile.d`, your changes are safe.

3.8 Package Management

Package management is handled by a shell script called `pacman` (a pun on Package Manager). Its options are somewhat similar to Red Hat's package manager RPM. It is possible to install and remove packages and to query the database of installed packages and files.

The binary packages used by `ttlinux` are basically `tar` archives compressed with the `bzip2` algorithm. You can download all the packages that normally come with the distribution from the downloads page on the `ttlinux` homepage. You may need to do that if you remove a package on a running `ttlinux` system and later want to reinstall it.

To install the package `bash-3.0-2.bin.tbz`, you would use the following command:

```
pacman -i bash-3.0-2.bin.tbz
```

You can also install a package from standard input, which can be useful for forming a pipe with the `wget` program. This allows you to install a package from the network without having to store it on the `ttlinux` system. For example, to download the `bash` package from the hypothetical `foo.org` web server, you would use:

```
wget http://foo.org/bash-3.0-2.bin.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` package, run

```
pacman -ql e2fsprogs
```

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

```
pacman -e isdn4k-utils
```

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.

The `pacman` script get its knowledge about installed packages and files from the directory `/var/log/packages`, where you can find one file per installed package. The file contains a listing of all files that belong to that particular package. `pacman` will not notice when you delete files by hand, it exclusively uses information found in `/var/log/packages`.

3.9 Using devfs (no longer supported)

Previous versions of `ttylinux` had support for running Linux kernels with the `devfs` device filesystem enabled. However, starting from the 2.6 kernel, `devfs` is considered obsolete by the main kernel developers. A replacement solution is called `udev` and runs completely in user space. As a consequence, I have dropped the minimal `devfs` support that was present in `ttylinux`. As of now, `udev` integration is not planned because I still want to support pre-2.6 kernels, but that may change in the future.

4 Contact

If you have 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:

Pascal Schmidt <ttylinux@ewetel.net>

See the file `AUTHORS` in the `ttylinux` distribution for a list of people who contributed ideas, bug reports, or code so far.

A How to create a bootable ttylinux CD

The following instructions are left here for reference purposes and will not work at all with a newer `ttylinux` with an 8M filesystem image size.

Please note: the following instructions are for putting together a 2.88 MB floppy image that can be put onto a bootable CD-ROM. This is a different approach to that used by the official

ttylinux CD-ROM version. The recipe below produces a smaller image than the official one, but the distribution installer will **not** work with the smaller images. There is a script called `mkttyiso` that is provided with the binary and source tarball versions of the distribution that can be used to put together the larger, installer-compatible CD-ROM images.

To follow the instructions, you need the following programs on your working system: `mkisofs`, `mkdosfs`, and `syslinux`. `mkdosfs` is usually found inside a package called `dosfstools`.

You will need the ttylinux filesystem image `rootfs.gz` and a Linux kernel image (the result of a `make zImage` or `make bzImage`). I assume you have these files in your current working directory and the kernel image file is called `vmlinuz`. The following instructions will only work for `root`, not normal users.

First, go through the rest of this user guide and do all the customization you want to do. Once ttylinux is burned onto the CD, you don't get any further chances to change anything.

Now we start preparing the CD image.

```
mkdir mnt
mkdir -p iso/boot
dd if=/dev/zero of=image bs=10k count=288
mkdosfs image
syslinux image
mount -o loop image mnt
cp vmlinuz mnt/linux
rdev mnt/linux /dev/ram0
cp rootfs.gz mnt
```

At this point, we have created an image file resembling a 2.88 megabyte floppy disk, `syslinux` has put a bootloader onto this image, we have mounted the image on the `mnt` directory and copied our kernel image and filesystem image files onto the disk image. We also told the kernel to expect the root filesystem on `/dev/ram0` when booting.

Now we need to create a `syslinux.cfg` configuration file on the disk image.

```
vi mnt/syslinux.cfg
```

In this file, place the line

```
append initrd=rootfs.gz
```

If you need to pass kernel parameters to your kernel, do this on the same append line. For example, if you need to pass `idebus=33`, your line should be

```
append initrd=rootfs.gz idebus=33
```

Now we are finished creating the disk image we need for making a bootable CD. Let's start putting together the ISO image.

```
umount mnt
cp image iso/boot/boot.img
```

The boot image itself is only 2.88 megabytes, so you will have extra space available. You can mount the CD after booting to get access to all the files you put on the ISO image in addition to the `boot.img` file. Simply put anything you want into the `iso` directory now. Once you are done with that, do:

```
cd iso
mkisofs -r -b boot/boot.img -c boot/boot.cat -o ../bootcd.iso .
cd ..
```

Congratulations, you finished the job of creating a bootable ISO image with ttylinux on it. All you need to do now is use some CD burning program to put the `bootcd.iso` image on a real CD. You could also save the `iso/boot/boot.img` file and use it to make bootable discs out of all the ISO images you create – all that is needed is about 3 megabytes of free space on the CD.