ROOT GNU/Linux v1.4 (Pommes)

A powerful all-purpose operating system.
John Eriksson <kohn at kth.se>

Manual version 0.9

17th December 2004

Contents

1 Introduction

This is a short introduction to the ROOT GNU/Linux system.

1.1 What is ROOT?

ROOT is a GNU/Linux operating system aiming for speed, simpleness and flexibility. It contains many high quality software packages for both server and workstation use.

The system is built mainly for experienced GNU/Linux or UNIX users. ROOT uses a simple text-based installation program (with menus). It uses simple tarball packages and BSD-style initscripts.

The system is distributed on one CD (700 MB) and is Pentium optimized. This means it requires an i586 (or compatible) processor, or newer.

ROOT uses KDE as its' desktop environment – only the GNOME *libraries* are included.

For more information, see the *Frequently asked questions* section.

1.2 License Information

ROOT is an open source project. This means, all source code written by me for ROOT is available under the GNU General Public License (GPL). The system contains lots of different packages with individual licenses. The most common license types are the GNU GPL and BSD-type licenses. Some non-free packages are also included in a seperate directory – for example *Macromedia Flash* and *RAR*.

The ROOT system comes with ABSOLUTELY NO WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. Use it at YOUR OWN RISK.

2 Installing ROOT

This section describes how to install the ROOT system. Remember that there is not any support available, so you'll have to manage this on your own. There could be help available on the Internet, however.

2.1 Supported Hardware

Here is a list of hardware supported by the ROOT GNU/Linux system. Most modern computers with Intel compatible CPU's will work.

IDE systems will more likely work than SCSI systems. If you have SCSI devices that must be supported during the installation, make sure your SCSI controller is supported by a driver in the *SCSI controllers* list.

2.1.1 Supported CPU's

All Intel Pentium (i586) compatible processors should work. This includes most newer Intel processors as well as many AMD processors.

2.1.2 Supported IDE controllers

All generic IDE controllers should work without problems.

2.1.3 Supported SCSI controllers

The SCSI controllers in the list below should work. However, it's not possible for me to test all of them. Therefore, most of these controllers havn't been tested.

The supported SCSI controllers are: 7000FASST, ACARD, Adaptec AHA152X/2825 / AHA1542 / AHA1740 / AIC7xxx AdvanSys, AM53/79C974 PCI, BusLogic, Generic NCR5380/53c400 Initio 9100U(W) / INI-A100U2W, NCR53c406a / NCR53C8XX, Simple 53c710 Symbios 53c416, Tekram DC390(T), Am53/79C974, UltraStor.

2.1.4 Supported keyboards

All generic PS/2 and USB keyboards should work without problems.

2.2 Installation Guide

This is a step-by-step guide to the installation process of the ROOT GNU/Linux operating system.

2.2.1 Get a ROOT GNU/Linux CD-ROM

If you don't own a ROOT CD-ROM, you must download the ISO image from a ROOT download server. This is currently the only way to obtain a CD. You need a fairly fast connection because the size of the ISO is about 700 MB.

When you have downloaded the latest ISO file, ensure your download was successful using the md5sum command on your GNU/Linux or UNIX system. The output of this command should be the same as the contents of the file rootlinux-x.x.md5sum (where x.x is the version number), found in the same directory as the ISO file on the server.

2.2.2 Make Boot Floppy (If needed)

If your computer does not support booting directly from the CDROM, you must make a boot floppy to be able to start the installer. You need one non-damaged floppy disk (1.44 or 2.88 MB).

In a Linux/UNIX system, mount your ROOT CD-ROM and change to the image/ directory on the CD. Then, as a priviligied user, type:

```
# dd if=boot.img of=/dev/fd0 bs=1024
```

In a DOS or Windows 9X system, use the program RAWRITE, available in the tools/ directory on the CDROM. In a Windows NT-based system (NT, 2000, XP), use RAWRITENT, available in the same directory. In the program, enter the path to the boot.img file and the destination drive (probably A:).

2.2.3 Before Installing

Please remember that ROOT comes with NO WARRANTIES AT ALL. If you have important data on your hard drive, make a backup before starting the installation. Nobody can be held responsible if your data gets erased.

2.2.4 Start the Installer

Insert the ROOT CD into your CD-ROM drive and (re)start your computer. If you're using a boot floppy, insert that one as well.

You should now see a welcome message starting with *"Welcome to ROOT GNU/Linux version 1.4 release 'Pommes'!"*

Press ENTER to start the installation from the first CDROM drive on your system. If you have several CDROM drives, and want to run from the second one, type:

```
install root=/dev/cdroms/cdrom1
```

Note that *cdrom1* is the second CDROM drive – *cdrom0* is the first. You can also boot from your third CDROM drive by replacing the '1' with a '2' and so on.

When the kernel has booted, the installer will start. You will see the message *Initializing ROOT GNU/Linux installation...* on your screen.

2.2.5 Keyboard Configuration

Select your preferred keymap from the list presented on your screen. The first two characters of the keymap generally means the country code for the keymap.

2.2.6 Partition Setup

Setup your partitions by choosing a program from the list. The supported programs are currently *cfdisk* and *fdisk*. cfdisk is menu-based and easy to use. fdisk is command-based, and harder to use – but much more powerful.

Start your selected program and create partitions if you need to. You must have at least one partition of the type Linux. You should also have a small partition (about 150 MB) of the type Linux Swap.

Don't forget to write your new partition table to the disk using the *write* function in cfdisk or using the command *w* in fdisk.

2.2.7 Activate Swap Partition

If you have created one or more Linux swap partitions, then the installer will detect the partitions and, for each partition, ask you if you want to use it as a swap partition. You probably want to answer yes to these questions. If you don't have a swap partition, you won't get this question.

2.2.8 Choose Target Partitions

A list of Linux partitions will be presented on your screen. Choose the partition you want to use as your root (/) filesysten, and press ENTER. Then select / as the mount point for the partition. Next, select your preferred filesystem. I recommend using either reiserfs or ext3.

If you want to have a seperate partition for your /home, /var, /usr or /opt, then select another partition from the list and select the mount point (for example /home). It is recommended to place your /home and /var directory on seperate partitions.

When you have added your partitions, press OK on the partition list to continue.

2.2.9 Package Selection

Now it's the time to choose what software packages you want to be installed to your new system. You first choose packages from the base category, then opt, x and so on.

If you are unsure if you need a package, don't uncheck it. ROOT has no dependency checking, so if you remove something required by another package, the other package won't work. However, unless you remove something extremely vital, you can install the package later.

2.2.10 Package Installation

The packages will now be installed. This will take some time.

2.2.11 Network Configuration

Select your hostname and domainname. If your computer is not on a domain, you can choose whatever you like.

If you are connected to a local network or Internet, then select your connection method from the list - either dynamic IP from a DHCP server or static IP (you have to enter your IP address yourself). If you are using a dial-up connection or no connection at all, choose loopback here.

If you didn't select loopback, the installer will search for a supported NIC (*Network Interface Card*) now.

2.2.12 Root Password Configuration

Set your root password here (and make it secure).

2.2.13 Boot Loader Configuration

If you want to install LILO (the boot loader for ROOT), then select yes here. You probably want to do so. To install to the suggested location (probably correct), just say yes to the question. Else, enter the correct device name.

You will also be prompted for special kernel parameters. If you want to use any such parameters, enter them on the line. The kernel will use these options when booting the system later. If you don't know what this is, or don't need any options, just press ENTER.

2.2.14 Time Configuration

Select weather your computer's hardware clock is set to local time or UTC time. Most computers are set to local time. If you have a Windows operating system on your computer, this is for sure the correct answer. If you don't know, choose local time.

2.2.15 Timezone Configuration

Select the correct timezone for your computer.

2.2.16 Restart Computer

The installation is now finished. Press ENTER to restart the computer. If you installed the boot loader, it should start when your computer boots up.

3 Configuring the System

This section describes some of the basic steps that should be done after an installation. This only includes ROOT-specific steps.

3.1 Adding Users

As you know, you shouldn't use the root account all the time. One of the first things you should do is to add a user to the system. You can do this using the *adduser* command. This will also install the default bash settings, and (if KDE is installed) the KDE settings. To add a user, type:

```
# adduser <user>
```

After this you need to set a password for the user. Do this by using the standard *passwd* command:

```
# passwd <user>
```

Now you can login as the new user.

3.2 Configuring X

ROOT uses the X.org X Window System. The configuration programs for this system are called *xorgconfig*, which is text-based, and *xorgcfg*, which is graphical. If you want to install your own X configuration file, it should be placed in /etc/X11/xorg.conf.

3.3 Setting Window Manager

Use the command *setwm* to set your desired window manager. Log in as a user and type:

```
$ setwm
```

This gives you a list of installed window managers. Choose one of them and then type:

```
$ setwm <window manager name>
```

Then you can start X using the *startx* command.

3.4 Loading Kernel Modules

Hardware like sound cards needs the appropriate kernel module loaded in order to work. Kernel modules are located in /lib/modules and are loaded using the standard *insmod* or *modprobe* commands.

If you want to load a module on boot, you should add the module's name to /etc/rc.d/rc.conf. If you want to load several modules, seperate the names by a space. Don't include the ".o" extension. Example:

```
MODULES="sis900 emu10k1"
```

3.5 Starting Services

Services in ROOT are handled using the *sstart* and *sstop* commands. These commands searches for a service file in /etc/rc.d/services. All official ROOT packages includes such a file. The sstart commands calls the start function of the service file and vice versa.

   If you want a service to be started automatically on boot, you should add the service's name to /etc/rc.d/rc.conf. Seperate several services by a space. Example:

```
 SERVICES="syslogd klogd inetd sshd cupsd"
```

4 The Package System

The package system used in ROOT is based on pkgutils from the CRUX Linux distribution. It is very simple, but elegant and powerful enough. A ROOT package has the filename *name#version-release.pkg.tar.gz*. This file only a gzipped tar archive of the files, not including any meta-data.

   pkgadd(8), pkgrm(8), pkginfo(8), and pkgmk(8) are the package management utilities. They are used to handle all you may want to do with packages.

4.1 Using the Package System

When a package is installed using pkgadd(8) a new record is added to the package database (located in /var/lib/pkg/db). Please note that the package system does not have any dependency checking. You must install all other packages that a single package may need manually.

   Install a package using the command pkgadd(8). The program requires at least one argument - the name of the package you want to install. For example:

```
# pkgadd mozilla#1.2.1-1.pkg.tar.gz
```

   If any file from the package is already installed on the system, the program will abort with an error message, showing the names of the conflicting files. Example:

```
# pkgadd mozilla#1.2.1-1.pkg.tar.gz
usr/bin/mozilla
usr/lib/mozilla/libgtkmozembed.so

pkgadd error: listed file(s) already installed (use -f to
ignore and overwrite)
```

   To force the installation (i.e. overwrite the conflicting files), use the option -f (or –force). For example:

```
# pkgadd -f mozilla#1.2.1-1.pkg.tar.gz
```

The package system only allowes a file to be owned by exactly one package. When forcing an installation the ownership of the conflicting files will be transferred to the package that is currently beeing installed. Directories can be owned by more than one package.

Please note that this option should only be used if you really know what you are doing. Two packages should never include the same files. If they do, it can be a sign that one of them is broken and contains too many files.

The package manager uses only the package filename to identify the package information such as name and version. If the package is named mozilla#1.2.1-1.pkg.tar.gz then it will be interpreted as mozilla version 1.2.1-1. If pkgadd(8) can't interpret the filename, then it will abort installation and print an error message.

4.1.1 Removing a Package

Removing a package is done by the program pkgrm(8). The program requires one argument, the name of the package to remove. Example:

```
# pkgrm bash
```

Note that this will remove all files owned by the package, without asking any questions. You don't want to remove the wrong packages, so make sure you spell the name right.

4.1.2 Upgrading a Package

Upgrading a package is done by using the program pkgadd(8) with the -u option. Example:

```
# pkgadd -u mozilla#1.2.1-1.pkg.tar.gz
```

This command will replace the previously installed mozilla package with the new one. If mozilla was not installed before, you'll get an error. Note that the program doesn't care about version numbers, so you can use the -u option to install an older version of a package.

The configuration file /etc/pkgadd.conf contains information on what to do when upgrading packages. The file contains rules, which is built out of three things: event, pattern and action. The only event currently supported is UPGRADE, and the action can be either YES or NO. More than one rule of the same event type is allowed, in which case the first rule will have the lowest priority and the last rule will have the highest priority. Example:

```
#
# /etc/pkgadd.conf: pkgadd(8) configuration
#
```

```
UPGRADE          ^etc/.*$                 NO
UPGRADE          ^var/log/.*$             NO
UPGRADE          ^etc/X11/.*$             YES
UPGRADE          ^etc/X11/XF86Config$     NO


# End of file
```

This will cause pkgadd(8) to upgrade anything in the directories /etc or /var/log, except files in /etc/X11, unless it is the file /etc/X11/XF86Config. The default rule is to upgrade everything.

Note! A pattern should never contain an initial '/' since you are referring to the files in the package, not the files on the disk.

If the program finds a file in the package that should not upgraded, it will be installed under /var/lib/pkg/rejected/, so that the user can decide what to do with them.

4.1.3 Querying the Package Database

Use pkginfo(8) to query the package database. The program has a few options. Please see the manual page for more information about each option.

5 ROOT Tools

The ROOT system comes with several ROOT-specific tools used for different purposes. This section describes their functions and usage. For more information about each tool, see it's corresponding manual page.

5.1 setwm

SetWM is a small utility to set the default window manager to start when X starts (for example with 'startx'). Typing 'setwm' without arguments will show a list of available window managers (the files found in /etc/X11/xinit/xinitrc.*).

When creating window manager packages for ROOT, be sure to include such a file with correct commands for starting the window manager.

5.2 sndtest

SndTest is a small utility to check if the sound card is properly configured. When you have loaded your sound card's modules, try running 'sndtest' and you should hear a sample sound (/usr/share/rl/system/rl.wav). This utility requires sox to be installed.

5.3 adduser

Adduser is a script to make it simple to install a new user. The program does just what it says, but it also installes the standard settings for the shell and so on. Generally use this program to install new users - do not use 'useradd'.

5.4 s[start,stop,restart]

These utilities are used to start/stop/restart services (daemons) on your computer. A service should have a service file included in /etc/rc.d/services to work. This file is a script which handles two arguments: start and stop. The script contains the lines needed to start or stop the service. The program srestart will first stop the service, then start it again.

5.5 gtk-update

This is a simple script to update the *GTK+* and *pango* configuration files. This may be needed after an upgrade or if something messed up the configuration. If GTK+ apps doesn't work, try running *gtk-update* once, as root.

5.6 dm

DM is short for decision maker. It outputs either *Yes.* or *No.* randomly. It should be used when you can't decide something yourself and need help ;-).

6 Frequently asked questions

This section contains answers to the most frequently asked questions about the ROOT system. This section has not yet been written.