Installing LinuxPPC−2000 Q4 on the IBM RS/6000 43P model 7248 HOWTO

Ingvar Hagelund

ingvar@linpro.no

Revision History

Revision 1.52	2001−08−29	Revised by: ih
Added a chapter on Linux 2.4. Fixed some dead and wrong links. Fixed a lot of typos.		
Revision 1.51	2001−04−06	Revised by: ih
Download site for LinuxPPC−2000 Q4 cd images		
Revision 1.50	2001−02−06	Revised by: ih
Now supports LinuxPPC−2000 Q4		
Revision 1.40	2000−12−14	Revised by: ih
Translated to SGML. This is the initial release for LDP		

This document describes the installation of LinuxPPC−2000 Q4 on the IBM RS/6000 43P model 7248 series.

Table of Contents

Table of Contents

Table of Contents

1. Introducion

This document describes how to install LinuxPPC on the IBM RS/6000 43P 7248 series, that is, the 43P−100, 43P−120 and 43P−133. It describes quite in detail anything to get one of these boxes from a non working stage to a networked X workstation. This relase covers LinuxPPC−2000 Q4. For older versions of LinuxPPC, please have a look at my homepage at http://users.linpro.no/ingvar/43p.

Some years ago I got a couple of old 7248s for free and I did not have any OS to run on them. So I gathered some bits and pieces from the net, and got it to install LinuxPPC−1999. Later, I found that a lot of other people may have the same problems that I had, so I wrote this document to help. It has been availble for some time in HTML form only. Now, it's moved to SGML and a part of the LDP.

1.1. Copyright Information and Legal stuff

Updated: August 27, 2001.

1.2. Disclaimer and scope

This document is made after own experiences on a 43P 7248−132. The things I did might or might not work for you. You are on your own. I take no responsibility whatsoever for any damage, loss or expenses because of something you might have done because this document said so. If you want to give me feedback on errors, typos, or anything that can make this document better, please feel free to contact me by sending an e−mail to <ingvar@linpro.no>

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies, that may of course be damaging to your system. Proceed with caution, and although this is highly unlikely, I do not take any responsibility for that.

This document is about installing LinuxPPC−2000 Q4 on the IBM RS/6000 43P, model 7248. The methods described in this document may or may not work on other machines or models. They may or may not work on other Linux distributions. Don't ask me about this, as I have not tested others. Look in the Section 13 for other resources on this. If you find that these instructions work on other models or distributions, please let me know, and I'll add that info here.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

Note that the 7248−132 and 43P−132 is two terms for the same model, and the same goes for 7248−133 and 43P−133. These four terms describes almost the exactly same model, with modifications so small, I have not been able to find them. You can safly assume that they are all the same machine.

If you feel that this document makes your life better, makes you glad and happy, or if you just are in a good mood, and have nothing to do, feel free to donate hardware, money, pizzas, e−mailed thankyous, postcards or anything to me. I can be reached at:

e−mail: <ingvar@linpro.no>
snail−mail: Ingvar Hagelund, Asperudlia 15, NO−1258 OSLO, NORWAY.

1.3. New versions

Version 1.52

- Added a chapter on Linux−2.4
- Fixed some wrong links and typos

Version 1.51

- Download site for LinuxPPC−2000 Q4 cd images

Version 1.50

- Updated to cover LinuxPPC−2000 Q4

Version 1.41:

- Made some small changes to the kernel chapter.

Version 1.40:

- Added a FAQ section.
- Reformatted howto to SGML.
- New copyright notices.
- Ready for bundling with the LPD.

If you are reading an offline version of this document, please note that an up to date HTML version can be found at http://users.linpro.no/ingvar/43p.

1.4. Credits

A lot of people have given me suggestions and help on these pages. I might have forgotten some of them, and if so, I apologize. Please send me a note to me at <`ingvar@linpro.no`>, and I'll list you here. Thanks to all of you, I could not have done this without you.

Ingvar

Here is list of people that have been helpful, in a completely unordered fashion :−)

Ingvar Hagelund

James Rooker

Mike McCammant

Alberto Varesio

Rolf Brudeseth

Ian Dale

Hollis R Blanchard

Linar Yusupov

Cort Dougan

Roger Bonussen

Rolf Zimmerli

Philippe Senot

John Roebuck

Jacopo Silva

Martin Espenschied

Dan Burcaw

www.linuxppc.com

Tor Arne Rein

Chien−Yu Chen

Wadamori Naoki

Arne Chr. Jørgensen

Doc Shipley

Thomas M. Nymand

Alberto Varesio

Pat Berge

Kazunori Aoshima

David Monro

Matt Porter

Olaf Hering

Xavier Piednoir

Steve Cornett

Greg Ferguson

1.5. Feedback

Feedback is most certainly welcome for this document. Without your submissions and input, this document wouldn't exist. Please send your additions, comments and criticisms to the following e−mail address : <ingvar@linpro.no>.

1.6. Translations

This document exists in English only. If you want to translate this document into an other language, please do so, just give me a note, and read the copyright notices above.

2. Overview

2.1. The IBM RS6000 43P 7248−133

This chapter contains a short overview over the 7248−133 and Linux for PowerPC. The data for the other 7248 models should not be too different from this.

The IBM RS/6000 43P model 7248−133 is a not extremely new PReP based PowerPC workstation which was produced from 1995 through 1997. The 43P series includes a lot of machines, both CHRP and PReP based. Our model has among other things this to offer:

- PReP architecture
- 132MHz PowerPC model 604 processor
- 512K synchronous L2 cache
- Max 192MB RAM (EDO)
- Integrated ncr53c810 SCSI−2 controller
- Integrated IDE controller
- Integrated AMD PCnet32 PCI ethernet adapter
- Integrated IBM E15 2MB graphics adapter based on the S3 Vision864 chip
- Integrated Crystal Audio cs4232 sound adapter
- 1.44MB Floppy disk drive
- IBM 8X SCSI CD−ROM drive
- Usually equipped with a 1.2MB IBM DPES 31080 SCSI hardisk

You can find more information at IBM's sales manual for this machine, found at this horrible URL

Linux has a native port to the PowerPC processor, and all official code is maintained in the main kernel tree. The main part of the port was done by Gary Thomas. The story of the port can be found here. For more information on running Linux on the PowerPC processor, check out the LinuxPPC homepage http://www.penguinppc.org.

2.2. What's the matter, why not use the original installation procedure?

The LinuxPPC−2000 Q4 distribution is made primary for the PowerMac Apple Macintosh. The installation programs has therefore no direct support for the 7248 nor the PReP architechture in general. But no worries, we'll make a few hacks, some manual configuration, and make it work anyway. Don't despair, this document will lead you through it, step by step.

2.3. An overview on what to do

Here is what we are going to do:

- Set up the hardware properly ("BIOS settings") with IBM's System Management Services (SMS)

- Get installation files from CD or FTP
- Make boot floppies
- Get the machine to boot and start the installation program
- Partition the hard drives
- Install the system
- Boot the system
- Set up networking
- Update the kernel and set up X
- Set up sound

3. Setting up the hardware with SMS

This chapter describes how to configure the system's Firmware to use the hardware properly. To do this we are going to use a software tool from IBM called System Management Services, SMS.

3.1. Where is the BIOS?

Lots of Linux users are familiar with the x86 platform, and immidiatly ask "What key may I press to access the BIOS". Well, on the 7248−133 it's not that easy, but it's not very difficult either. This is a real UNIX machine, and real UNIX machines don't have a BIOS. They have some sort of Firmware instead. There is not much to yell about, 'cause to the users, it's almost the same thing. Firmware often seem to have lots of bugs in them, just like the BIOSes in the PC world. They often are more selective on what kind of hardware they support too. For hardware system management on our 7248, IBM has a software program to be booted from floppy disk, called "SMS" − "System Management Services". With this you can access the Firmware. You can get floppy images and MS−DOS programs for creating those here. You might also want to upgrade your Firmware. Check the README files on the link about this.

3.2. How to use the SMS

To boot the SMS, turn off the machine, insert the SMS floppy, push the power button and press the F4 key while the system check icons pop up in the bottom of the screen. (If you prefer a text based interface, try F2 instead.) Basic usage is outside the scope of this document, but it's not very difficult.

3.3. What settings to use

You should have a little look over your system to see whats actually there. Then you should check the boot sequence, as it's quite important to make your system boot right. The sequence should be: (You guessed it) Floppy disk −> CD−player −> first SCSI harddisk −> second SCSI harddisk −> etc.

3.4. More info about the hardware

You might find good references in IBM's salesmanual for our model.

There are also some reference manuals in pdf format for free download from the IBM RS/6000 pages that might come handy.

4. Get the installation files

This chapter describes how to get your hands on a copy of the LinuxPPC−2000 Q4 operating system.

4.1. Buying a CD

The easiest way to get all the files you need is perhaps just to buy an official CD set. You may order the CD set directly from http://order.linuxppc.com or you may try to find a local reseller, linuxppc.com has a list here.

4.2. Download CD images over FTP

The other way around is to download preformatted CD images, and burn them out yourself. You can download installation CD images for example from Linuxiso.org. You should download the LinuxPPC−2000 Q4 install image and the LinuxPPC−2000 Q4 software image. There are some interesting readme files on the download sites on burning the files to a CD−ROM.

To make a CD, you need some CD burning software package. This is outside the scope of this document, allthough any good CD burning software should be able to do it. Remark that the images are in HFS format, and NOT the usual iso9660 format.

4.3. Network installation

Other Linux distributions have often support for direct installation via a network connection. This is not currently supported in LinuxPPC−2000 Q4, at least not on these machines, yet.

5. Make boot floppies

In this chapter we will learn how to make useable boot floppies from files downloaded from the Internet.

5.1. What floppies to make

To install LinuxPPC−2000 Q4 on the 7248 you need three floppies: One bootfloppy, and two ramdisk floppies.

The bootfloppy is a precompiled Linux kernel image called carolina_bootimage_2.2.18_with_fb, and it can be found in the images directory on my homepage.

The ramdisk floppies are in the ramdisk.image.gz file found in the install directory on a LinuxPPC ftp mirror or on the CD. But it would be easier, at least for ms−dos users to download 1.44MB sliced images, also from my images directory over http.

If you use Netscape or another web browser to download the files, you should check that the sizes of the downloaded files are correct. Some versions of Netscape tend to uncompress compressed files, and we want to keep them compressed. This yields especially for the ramdisk.image.gz file(s). If strange freezes or other things happens at boot time, try using another program for downloading the files, like wget or lynx.

5.2. How to make the bootfloppies

Use always errorfree 1.44MB floppies for these images. The commands shown here is for a working Linux system. They might work on other UNIX systems as well. On some systems you may have to be root to write directly to the floppy drive. In those cases, just su root before issuing the commands.

For MS−DOS, you may use the rawrite util. You can download rawrite from several places, for example a RedHat mirror as ftp://ftp.freesoftware.com. More information on how to use rawrite here.

To make the bootfloppy, insert a floppy in the drive, cd to the directory containing the bootfloppy image and issue the command:

```
          dd if=carolina_bootimage_2.2.18_with_fb of=/dev/fd0 bs=36b conv=sync
```

Label the disk "Boot floppy" or whatever you like.

To make the first ramdisk floppy, insert a floppy in the drive, cd to the directory containing the ramdisk image, and issue the command:

```
       dd if=ramdisk−2000−Q4−floppy−image.1 of=/dev/fd0 bs=36b conv=sync
```

(If you use the ramdisk.image.gz from the CD or a ftp mirror, try dd if=ramdisk.image.gz of=/dev/fd0 bs=36b conv=sync count=80) Label the disk "Ramdisk root floppy (#1)" or whatever you like.

To make the second ramdisk floppy, insert a floppy in the drive, cd to the directory containing the ramdisk image, and issue the command

```
        dd if=ramdisk−2000−Q4−floppy−image.2 of=/dev/fd0 bs=36b conv=sync
```

(If you use the ramdisk.image.gz from the CD or a ftp mirror, try dd if=ramdisk.image.gz of=/dev/fd0 bs=36b conv=sync skip=80) Label the disk "Ramdisk floppy #2" or whatever you like.

6. Boot the machine and start the installation program

In this chapter we will find out how to get the installation program up and running.

6.1. Boot the machine

To boot the 7248, just insert the bootdisk and switch on the machine. Insert the installation CD at the PowerPC splash screen. If it won't boot off the floppy drive, check SMS settings (Section 3), and try to force a floppy boot by pressing F5 (or F6) at the bootscreen while the check icons pop up in the bottom of the screen. After a while, the screen blanks out, and Linux will boot. At this point, you can shout a little "hooray" for yourself, if there are not too many in the room, and Tux, the Linux Penguin will show up in the upper left corner of the screen. Insert the ramdisk root floppy and ramdisk floppy #2 when prompted.

After some seconds, the installation program will complain about a lot of things. It can't find a usb mouse, it can't start the Xpmac X−server, it can't open the local display, and that kind of things. Don't worry, we'll take care of this in a minute. Just press enter, and you'll be thrown to a standard root bash prompt. For those of us who have used Unix systems earlier, this is enough to give good vibrations. For all others: Don't Panic [tm].

6.2. Make some hacks

To be able to start the installation program, we have to change the standard PowerMac setup to something that is useable for the 7248. This is what we are going to do in this section. Remember that all commands are case sensitive.

First, let's set up mouse, to make them work correctly with X. Issue these commands:

```
          rm −f /dev/mouse
          ln −s /dev/psaux /dev/mouse
```

Done! Now, that wasn't too difficult, was it? Let's continue with some X configuration. Issue these commands:

```
        cd /etc
        mv X11/XF86Config X11/XF86Config.old
        cp XF86Config.PReP X11/XF86Config
        rm −f /etc/X11/X
        ln −s /usr/X11R6/bin/XF68_FBDev /etc/X11/X
```

This XF86Config.PReP file is not complete, it so we have to fix it. The easiest way to do this is to run Red Hat's Xonfigurator program

```
        Xconfigurator
```

and follow the instructions on the screen. The usage of Xconfigurator is beyond the scope of this document, but it's quite easy. If you need help, try the Xconfigurator section in the Red Hat Installation guide. Xconfigurator should figure out that we use the IBM E15 frame buffer device by itself. Note that the frame buffer device has only support for 256 colors, so only 8 bit mode should be selected. Follow the on−screen instructions and enter values that match your Screen. Resolutions of 800x600 or 1024x768 should be allright.

Skip the test at the end. It will fail.

If you try it ouy, you will maybe conclude that the X−server does not work. Again, Don't Panic [tm]. The reason for this is that Xconfigurator presumes you have an X font server running. You don't, so we have to add Font Paths manually. Edit the configuration with

```
        vi /etc/X11/XF86Config
```

(What? You don't like vi? Read this to get started.)

Look for the line which contains the FontPath. Comment out (that is: Put a '#' in front of) the existing line containing something like "unix/:7100", and add these lines instead.

```
        FontPath    "/usr/lib/X11/fonts/misc/:unscaled"
        FontPath    "/usr/lib/X11/fonts/cyrillic/:unscaled"
        FontPath    "/usr/lib/X11/fonts/100dpi/:unscaled"
        FontPath    "/usr/lib/X11/fonts/75dpi/:unscaled"
        FontPath    "/usr/lib/X11/fonts/Speedo/"
        FontPath    "/usr/lib/X11/fonts/Type1/"
        FontPath    "/usr/lib/X11/fonts/misc/"
        FontPath    "/usr/lib/X11/fonts/cyrillic/"
        FontPath    "/usr/lib/X11/fonts/100dpi/"
        FontPath    "/usr/lib/X11/fonts/75dpi/"
```

You are now ready to start the installation program. Issue this command, and go for it:

```
        start_installer
```

7. Install the system

In this chapter we will get used to the LinuxPPC−2000 Q4 installation environment, partition the harddisk(s) and install the operating system packages.

7.1. The LinuxPPC−2000 Q4 installer

So, now we're actually getting somewhere! At least, so it looks. This looks really cool, doesn't it? Press the right mouse button on the background on the screen. What a cute little menu! Notice that you can change the colors of the screen with the *Styles* option. This is the first installation program I've ever seen with changeable colors :−) Note that you can move the windows on the screen around by clicking and dragging on the blue top or right borders. To bring a window to the front, try clicking on it. Select *xterm* on the Blackbox (background) menu. This will bring up a command line terminal window which we will use to start some programs that can do what the installation program can't.

Behold! There are even two (!) installation programs. Unfortunately, they don't work. In the xterm window, kill the installation programs. We have to do some more hand−work.

```
        killall install_helper
```

7.2. Partition the harddisk(s)

The first thing to do is to partition the hard disk(s). Since we have brutally killed the installation program, THERE IS REALLY NO WAY TO DO THIS. I have framed you all the time. Just shut down the machine and have a beer. Then visit some friends.

Just kidding.

In the xterm window, issue the command

```
        fdisk /dev/sda
```

This will start the good old fdisk program. Note: *This program will wipe away anything on the disks in the machine.* If there are something on the disks that you want to protect, exit the program by hitting *Q* and press Enter. If you like a menu driven program, we'll start using cfdisk in a minute. For an overview on the fdisk commands, hit *?*. To view the present partition scheme, hit *P*. If there are lots of unknown AIX information there, hit *D*, and select 1, *D* again and 2, and continue all the way up to 5 to be sure all old AIX partitions are wiped away. Write the changes to the disk by hitting *W*, and quit the program by hitting *Q*. If there are no AIX partitions on the disks, just quit with the *Q* command. If you have more than one harddisk in the machine, use the commands fdisk /dev/sdb for the second harddisk, fdisk /dev/sdc for third and so on, and repeat the steps from the first disk.

You could of course use fdisk to make the new partitions too, but I prefer a more user friendly solution. At the shell prompt, type

```
        cfdisk /dev/sda
```

To start the cfdisk program on the first harddisk. Change to sdb and sdc, and so on for more harddisks. Basic usage of cfdisk is outside the scope of this document, but I have written a little starter. You can read it in the Section 16.

You should have at least these partitions:

- A boot partition. It should be the first primary partition on one of the SCSI drives, preferably the first (this naming the partition to sda1). It must have type PReP boot (type 41), and must be large enough to hold a compressed Linux kernel image (zImage). Something like 5−10MB should do.
- A swap partition. It can be either a primary or a logical partition on any drive. Any size will do, but a guide may be twice the size of you physical RAM. I have 64MB RAM, so I have a swap partition on 128MB. Remember: This is only a guide.
- A system root partition. It can be either a primary or a logical partition, and it should be big enough to hold the main parts of the installation. You should write down the device (disk and partition number, like sda5, for example) for your system's root partition. You will need it later. The easy way is just to use the rest of the harddisk space for this. If you have several harddisks, big drives or special requirements on safety and other things, you should consider to make own partitions for /home, /usr, /usr/local, and other parts of the system. Details on this is outside the scope of this document, but there is a good discussion on this in the Linux Partition HOWTO.

It is a *VERY* good tip to find a piece of paper and write down which partitions you have made, what you want to use them to, and where you want to mount them. You will need this information later. When you have finished partitioning your disks.

7.3. Mount the partitions

So, we have some nice partitions. Let's use them. We have to make filesystems on them, and then mount them to our existing installation filesystem to be able to copy any files to them. Now, I HOPE you have written down what partitions to use where. First mount the root partition:

```
        mkdir /mnt/install
        mke2fs /dev/sdxy
        mount /dev/sdxy /mnt/install
```

Of course, x and y are disk and partition number, as you have written down. I use sda5 for my root partition, so I use the commands 'mke2fs /dev/sda5' and 'mount /dev/sda5'.

If you have more partitions, mount them in the same way. I have a 1GB partition at sdb1 which I want to use for the /usr system. So I write this:

```
       mkdir /install/usr
       mke2fs /dev/sdb1
       mount /dev/sdb1 /install/usr
```

You may have other partitions and also other mount points, like /usr, /tmp, /home, /var and other stuff. But I guess you've got the point now.

7.4. Select and install packages

You have now cleared the first stage, and get 250 bonus points. Congratulations. If the installation program had worked, you had got no points at all, so be happy. The next stage is to install packages. Here we can finally use some semi−automatic programs. In the xterm window, type this:

```
        xupgrade --install --debug --path_to_root /mnt/install
```

A new window will pop up. Select packages after your own will. What packages should I select, you say. Oh, please! I have no idea how you're going to use the machine :−). Press Install. Aaah. There it goes! After some minutes (not to say hours, it feels like that, doesn't it?) the install will be complete.

7.5. Make a root password and exit

Before we finalize the installation, it's a cool thing to have a root password. Enter your new installation, and make a funny, not−easy−to−guess password like this:

```
        chroot /mnt/install
        passwd
        (You will be prompted for the new password twice)
```

You are still inside the freshly installed system (the chroot command does that magic). Now, let's fix some other stuff while we're at it. The mouse and the keyboard are not right. Same goes for the timezone. If you are an experienced user you may want to add even more configuration at this point.

```
        rm −f /dev/mouse
        ln −s /dev/psaux /dev/mouse
        kbdconfig
        timeconfig
```

Next, you may find that the /dev/cdrom link is not working properly, it points to itself. It should point to /dev/scd0, so let's fix this. Issue these commands:

```
       rm −f /dev/cdrom
       ln −s /dev/scd0 /dev/cdrom
```

You should now be able to mount the cdrom using /dev/cdrom. Thanks to Thomas M. Nymand for this tip.

Now, you should be all ready. Issue the command

```
       exit
```

to exit the installed system and return to the intallation system.

Finally, shutdown the system, remove the CD and floppy disk, and take a break.

```
        reboot
```

8. Boot the machine

In this chapter we will boot our fresh installed system for the first time, and make some small custom changes to make it work as it should

You are done! If everything worked, you should now be able to reboot you're system, and start LinuxPPC−2000 Q4 for the first time. Reinsert the boot floppy disk in the floppy drive, and switch the machine on again. If it won't boot, try to hit F5 at the splash screen while the system check icons pop up in the bottom of the screen. At the boot prompt, ("Linux/PPC load:") you must add a boot parameter to make the system find your root partition. (That's usually the main system partition.) Press backspace to remove what's already there, and add something like this:

```
       root=/dev/sda5
```

I use sda5 as sda5 is where I have installed my root partition, that is, the partition mounted at "/". You might have something different, and you should have written it down when you partitioned you harddisk(s). You did, didn't you?

The system should boot up, and after a minute or so, greet you with a login prompt. Congratulations, you have installed LinuxPPC−2000 Q4 on your computer! From here, you have to know how to use linux. This is absolutely outside the scope of this document, but if you are a complete newbie, you could for example check out Linux Administration Made Easy by Steve Frampton, and start at chapter 6, since you've already got your system up.

If you are not an US citizen, you should look over the X configuration file to get your local keyboard. Use a text editor like vi or pico, just like we did to get the installation system to work. Try

```
       vi /etc/X11/XF86Config
```

and scroll down to the Keyboard section. Check that you have settings that suits you. Edit as you wish. You may also want to remove LinuxPPC's annoying attempt to autoconfigurate X for you when the machine boots into runlevel 5.

```
       rm −f /etc/rc.d/rc5.d/*Xautoconfig
```

The next time you power−cycle the box, you must again boot from the floppy. The bootloader arguments at the boot prompt ("Linux/PPC Load:") should be still be something like this:

```
       root=/dev/sda5
```

Where of course, you may have something different from sda5, according to where your root partition is.

This is the time to install the rest of the system, with all the packages that are on the software CD. Login as root, open a terminal window, insert the CD, mount it, and start the package upgrade program:

```
      mount −t HFS −o ro /dev/scd0 /mnt/cdrom
      xupgrade
```

9. Set up networking

In this chapter we will set up TCP/IP networking on the integrated ethernet adapter

9.1. Support for the integrated ethernet adapter

The 7248−133 is equipped with an internal AMD PCnet 32 PCI ethernet adapter. There is support for this adapter in the Linux kernel source, and if you installed from my bootimage, you should have support for it in your kernel already. Otherwise, you can compile a new kernel. The bootimage described in Section 5 has support for the adapter too.

9.2. Configuration files

The configuration files to set up TCP/IP−networking over ethernet on this box is the same manual configuration as on any other RedHat−like Linux box elsewhere, but here, we do it manually, without any tools other than your favourite text editor. This only covers basic installation, and NO EXTRA SECURITY AT ALL. There are plenty of other, and much better sources about this topic than I can put up here. The files to edit are found in the /etc directory and its subdirectories. To edit the files, you should be logged in as root.

9.3. Set IP address and netmask

To set the IP address and netmask, make a file

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

That looks like this:

```
        DEVICE="eth0"
        ONBOOT="yes"
        IPADDR="111.122.133.144"
        NETMASK="255.255.255.0"
```

Of course, replace the numbers in the IPADDR and NETMASK fields with the ones you should use. If you don't know, ask your system administrator.

9.4. Set hostname and gateway

To set the hostname and gateway, edit the file

```
        /etc/sysconfig/network
```

so it looks like this:

```
        NETWORKING=yes
        FORWARD_IPV4="yes"
        HOSTNAME=myhost.mydomain
        GATEWAY="122.133.144.155"
```

Of course, replace the values in the HOSTNAME and GATEWAY fields with the ones you should use. If you don't know, ask your system administrator.

9.5. Nameserver specification

To specify a nameserver, edit the file

```
        /etc/resolv.conf
```

so it looks like this:

```
        search mydomain
        nameserver 133.144.155.166
```

Of course, replace the values in the search and nameserver fields with the ones you should use. If you don't know, ask your system administrator.

9.6. The hosts file

Finally, edit the the file

```
        /etc/hosts
```

so it looks like this:

```
        127.0.0.1       localhost       localhost.localdomain
        111.122.133.144 myhost.mydomain myhost loghost
```

The last line should suit the values you added previously as the system's *IP−address* and *hostname*.

9.7. Restart the network

This should be everything you need to get started. Now restart networking with the command

```
        /etc/rc.d/init.d/network restart
```

And you should be all up. Note that there is a bug somewhere that makes the netcard freeze the whole system when shutting down eth0. According to Martin Espenschied, this is a known issue, and can be fixed. When I know how, I'll put the information here. Till then, remember to *sync* your system before shutting down, and you should not miss any information. Remember my disclaimer in Section 1.2 though.

An ugly hack to resolve this might just be to *NOT* to shut down eth0 at shutdown/reboot. This can be done by editing the file /etc/rc.d/init.d/network, and in the stop) case, just comment away the ifdown command, and add a phoney command, like this:

```
        action "Shutting down interface $i" echo does_nothing # ./ifdown $i boot
```

Note that this is a *rather ugly* hack, that won't actually solve the problem, it just hides it. The network won't go properly down until you reboot the machine. This is actually the "stupid Windows" way. I really hope that somebody have a better fix on this later. Thanks to Doc Shipley for this tip.

10. Update the kernel and tune the X Window system

In this chapter we will download the Linux kernel sources, add a few patches, and compile and install our own custom kernel on the harddisk. When we have successfully accomplished this exciting event, we don't have to boot from the installation boot floppy anymore.

10.1. Why update the kernel

The kernel we have used this far is a 2.2.18 kernel that I have provided. This is a quite stable and good kernel, but it's not sure that it's what you want. You should make your own kernel so you know what patches you need and what modules you can install when you really need them. Here are links to all sources and patches, and a step by step guide to compile your new kernel.

As for the installation we just have done, X does work, but only when you boot from the boot floppy. Would it not have been nice to be able to boot directly off the hard disk − and by the way, what about sound support? Let's set up this together as quickly as possible.

Before starting downloading files and compiling the kernel, check that you have these packages installed:

- make
- gcc
- cpp
- glibc
- glibc−devel
- ncurses−devel
- kernel−headers
- This list should probably be longer

10.2. What files to download

We'll use the 2.2.18 kernel sources. The 2.2.19 kernel is availble, but it has some problems with compiling on ppc, so we'll stay with 2.2.18. To add better support for our Carolina motherboard, well add two patches from David Monro's Carolina page. This may sound complicated, but believe me, it's not. Follow the steps below here, and we'll get you up in an hour or so.

We're going to use the standard place for linux kernels, that is /usr/src/ There might be a linux directory there already, so let's rename it:

```
        mv /usr/src/linux /usr/src/linux.old
```

When downloading the kernel source and patches, place all files in /usr/src.

Begin with the linux−2.2.18.tar.gz source from the /pub/linux/kernel/v2.2 directory at ftp.kernel.org or a local mirror (recommended for faster download).

Then, get the Carolina patch and the IBM E15 frame buffer patch from David Monro's Carolina page at http://www.amberdata.demon.co.uk/carolina

10.3. Unpacking and patching the source

The next step is to unpack and patch the source files you just downloaded. cd to the directory where you downloaded the files, and unpack the kernel source by issuing this command:

```
        tar xzvf linux−2.2.18.tar.gz
```

(On some mirrors there are only bz2 files availble. If this is the issue, use xvIf instead of xvzf.)

This will make a directory called "linux" with all the kernel source. Now we shall add the patches. If the patches have the extentions "gz", unpack them with gunzip like this:

```
        gunzip carolina.diff.gz
        gunzip ibm_e15fb.diff.gz
```

Now add the carolina patch by issuing the commands:

```
        mv linux new
        patch −p0 <carolina.diff
```

Last, add the IBM E15 patch by issuing the command:

```
        patch −p0 <ibm_e15fb.diff
```

For a clean compile later, rename the kernel tree to its original name by issuing this command:

```
        mv new linux
```

If you get through these steps succesfully, the kernel source is now ready for configuring.

10.4. Configuring the kernel

The details of configuring the kernel are way outside the scope of this document. To get more help, try The Kernel HOWTO. Here, we'll just cover the basics to get a working kernel. Download my kernel config file into the linux directory in the kernel tree top directory.

In the directory containing the linux kernel tree, start the configuration program by issuing these commands:

```
        cd linux
        make menuconfig
```

Select the almost last option *Load an Alternate Configuration File*. Press backspace to remove the text in the box, type *ingvar.config* instead, and press enter. This config file has support for most you'll need on this box, including sound and the frame buffer device to run X. This should be enough to get the kernel to compile, but while you're in there, check over the different menus to get to know your possibilities. Don't be afraid to change anything. You can always load the configuration file again. A thing you really should check, and possibly change, is the *Initial kernel command string*, located in the *General setup* submenu. Change the value of the root device to the device where your root (/) filesystem is mounted. If unsure, press *ALT+F2*, log in and try the command *mount*. Press *ALT+F1* to get back again (supposed that you are on virtual terminal 1). Also check your own preferences on filesystems and other stuff you might have special need for or interest in. If you for some reason don't want (or need) to run X, eg. because the console becomes much slower, remove support for the IBM E15 frame buffer in the Console drivers submenu.

In the main menu, save your new configuration for later use, if you like, and select exit. Answer *Yes* to the question about saving the changes. You are now ready to compile the kernel.

10.5. Compiling and installing the kernel

To compile the kernel is quite straightforward. If you're not in there already, enter the linux directory in the top level of the kernel tree, and issue this command series:

```
        make dep && make clean && make zImage && make modules && make modules_install
```

Now, go and make yourself a cup of tea or coffee or have a beer or a bible study for some minutes. This can take some time. If everything is alright, you'll finally have a kernel to install when the text stops scrolling after maybe as much as 20 minutes, depending on your kernel configuration.

Now it's time to install your fresh−baked kernel. On RedHat−like systems like LinuxPPC, we keep kernels and their setupfiles in the /boot directory.

```
        cp /usr/src/linux/arch/ppc/boot/zImage /boot/vmlinuz−2.2.18
        cp /usr/src/linux/System.map /boot/System.map−2.2.18
```

Assuming that sda1 is your boot device, install the new kernel by issuing this command:

```
        dd if=/boot/zimage−2.2.18 of=/dev/sda1
```

We also want the System map link to point to our new System.map. Issue these commands:

```
        mv /boot/System.map /boot/System.map.old
        ln −s /boot/System.map−2.2.18 /boot/System.map
```

There! You are ready to reboot. Light some candles, remove the boot disk, and issue the command:

```
        reboot
```

If everything works out, the system will go down nicely, and (assuming you compiled in support for the E15 frame buffer) come up again with Tux, the Linux Penguin Mascot smiling to you while booting.

10.6. Set up X

In this chapter we will set up X, if it does not work properly already.

To set up X, check that you have installed the following packages:

- XFree86
- XFree86−FBDev
- XFree86−100dpi−fonts and/or XFree86−75dpi−fonts

- Xconfigurator
- Maybe more packages here...

Then as root, just run the command:

```
        Xconfigurator
```

And it will set up X for you. The usage of Xconfigurator is beyond the scope of this document, but it's quite easy. If you need help, try the Xconfigurator section in the Red Hat Installation guide.

11. Set up sound

In this chapter we will fix some sparkling sound from the built−in sound adapter.

11.1. Support for the integrated sound adapter

The 7248 is equipped with an internal Crystal Audio cs4232 sound adapter. There is support for this adapter in the Linux kernel source, but this is not compiled into the standard installed kernel. So, we have to compile a new kernel. If you are a total beginner on compiling kernels, you should read the excellent kernel howto on how to get started. Download the kernel source from ftp.kernel.org or a mirror. I tested this with kernel 2.2.10 and 2.2.16.

11.2. Configure and compile the kernel

You should configure the kernel as you wish. If you did follow the steps in the last chapter, you have allready done this, and can skip to the next section. If not, follow the steps as described in the last chapter .

When configuring the kernel, check that the settings for sound are correct. They should look like this:

- Sound: *Y*
- OSS sound modules: *Y*
- Generic OPL2/OPL3 FM synthesizer support: *M*
- Support for Crystal CS4232 based (PnP) cards: *M*

And check *N* for everything else in the sound section.

11.3. Configuring the modules

Before rebooting we must send some parameters to configure the modules at boot time. This is done in /etc/conf.modules . Edit this file, and add the following lines:

```
        alias sound cs4232
        pre−install sound /sbin/insmod sound dmabuf=1
        alias midi opl3
        options opl3 io=0x388
        options cs4232 io=0x534 irq=5 dma=1 dma2=0 mpuio=0x330 mpuirq=5
```

11.4. Reboot and pray

Now, just reboot and pray. If nothing works, you have done something unwise when configuring the kernel. Boot from a floppy and try again. If everything works, you should for example be able to play an mp3 file using the mpg123 player from the console. (And yes, *of course* you might have to install the mpg123 package first.)

Note: I have not been able to get midi (the opl3 module) working. If anybody has any tips on this, please let me know.

12. Linux−2.4

In this chapter we will see how to install the latest and greatest (and most unstable) kernel. Linux version 2.4 has been around for some months, and there is active development also on the PowerPC branch. To get a working kernel, we'll have to use a development tree with a lot of patches that hasn't got into Linus' official source yet. To state this explicitly: You may NOT use the official Linux 2.4 source tree, at least not without third party patches.

12.1. Get the 2.4 source

First we must get the working 2.4 source. To get this, we'll use the rsync tool, so check that you have it installed. Some nice people have set up an rsync server of the BitKeeper Linux/PPC development tree at source.mvista.com. Thanks so very much to them, remember them in your heart and prayers. Warning: Don't do this over a low−end link, like a modem. It'll take forever. So, let's rock and roll. Issue these commands:

```
        cd /usr/src
        mv linux linux.old
        mkdir linux−2.4
        ln −s linux−2.4 linux
        cd linux
        rsync −avz −−delete source.mvista.com::linuxppc_2_4_devel .
        chown −R root.root .
```

Note that you'll probably have to wait for some minutes to rsync over all the sources. This is normal.

12.2. Compile the 2.4 kernel

All right, so we have the source. Now, let's do some configuring, and then compile a brand new kernel.

```
        make ibmchrp_config
        make menuconfig
```

Note that we don't need to apply any pathces as most of the patches mentioned earlier are merged into this tree. Now configure the kernel as you like. Most of the stuff mentioned earlier should work. The very exception is the frame buffer for the graphics card, so don't expect a graphical workstation with 2.4, at least not yet, unlesss, though, someone gets a vanilla graphics card working with the usual XFree86 drivers. I haven't tested this yet.

All done? Good. Now issue the usual command series to build the kernel:

```
         make dep && make clean && make zImage && make modules && make modules_install
```

12.3. Install the 2.4 kernel

To install the 2.4 kernel, we use the same method as with the 2.2 kernel. Note that the zImage has changed name and location in the source tree.

```
        cp arch/ppc/boot/images/zImage.prep /boot/vmlinuz−2.4.9
        cp System.map /boot/System.map−2.4.9
        dd if=/boot/vmlinuz−2.4.9 of=/dev/sda1
```

Note that these commands are for a 2.4.9 kernel. If your kernel source is newer, it's good behaviour to use it's correct version and patchlevel in the name of the kernel image. You can find the correct kernel version and patchlevel with the command

```
        head Makefile
```

Now again: Reboot and pray. If all goes well, you'll be up running in a few seconds. If it doesn't work, just reboot with the floppy disk as described earlier.

13. Resources

In this chapter there is a list of resources which include enough reading to make us experts in the field in record−time.

13.1. Other resources on Linux/PPC and 43P boxes

- penguinppc.org our home for Linux on the PowerPC.
- Linux/PPC PReP page
- Linux/PPC CHRP page
- Kernel patches, utilities and hints for IBM PPC workstations with Carolina motherboard (like the 7248) by David Monro. (Including frame buffer support to actually get X up and working!)
- PowerPC Linux project pages at sourceforge.net
- IBM's Linux on PPC project pages
- Hollis Blanchard's page for installing Linux on the PowerSeries 850. Add this to David's and my pages, and you should be able to get LinuxPPC−2000 Q4 up on the 850 too :−)
- The Linuxppc−1999 installation instructions for PReP Simple instructions from the ftp site
- More info about Linux on PReP machines by Kazunori Aoshima
- A newbie's encounter with LinuxPPC is a document which describes a newbie user's results and experiences on installing linuxppc using own experience and this document. Thanks to Arne C. Jørgensen for this document.
- A very short view on how to set up LinuxPPC on the 7248 the quick and easy way. Also by Arne C. Jørgensen.
- The linuxppc mailing lists. Especially interesting is of course the workstation list.
- Some kernel developpment stuff
- IBM Sales Manual for the 7248−133 (Detailed description)
- Description on 7248 microcode from IBM
- Year 2000 info from IBM
- Order documentation for the 7248 from IBM
- Online books for 43P from IBM, not the 7248, unfortunately.
- PReP specification from IBM
- Cross Development for Linux/PPC from i386 by Matt Porter

13.2. Installing other flavours of Linux on the 7248

- Installation instructions for Debian GNU/Linux on PReP machines. (Note: The bootdisk included in the Debian distro does not work on the 7248−133 because of a SCSI driver bug. You might use the bootdisk described in Section 5 instead. It works.)
- Installation instructions for Yellow Dog Linux on RS/6000: Yellow Dog Linux claims to support some, but not all, 43Ps. The 7248−133 is not on the list, but some ideas from this document, including the bootdisk mentioned in Section 5 does actually work.
- SuSE's ppc pages. SuSE does include support for PowerPC in their newest version.
- NetBSD is working on a port to PReP PowerPCs. (Yes, I know. NetBSD is not a Linux flavour, but I thought I might mention it here anyway.) There is a a special page for our PReP architecture where the 7248 is mentioned explicit.

14. Todo

There are some things that might be added to this document. If you have comments, things to add or want to help, please send an e−mail to <ingvar@linpro.no>

- Get rid of that annoying freeze at eth0 shutdown. This should be a part of the networking chapter. If you know something about this, please let me know.
- Dual Boot with AIX. We need some kind of a bootloader. Does such a thing exist? We know quik does not work on the PReP boxes.
- Get midi to work. Is this possible?

15. Frequently Asked Questions

In this final chapter I've included som frequently asked questions. This list should probably be much longer. Please let me know if you have something to add.

15.1. XF68 or XF86

What is right, XF68 or XF86?

I have got a lot of questions conserning the name of the X−server in the installation program. I have called it "XF68_FBDev". On some CDs the server has got another name, "XF86_FBDev". The reason for this naming convention and confusion is purely historical. The Linux Frame Buffer Device was first developped on m68k Macintoshes, and the XFree86 server for the device was hence called XF68_FBDev. Later on the Frame Buffer Device was ported to other platforms like the x86 clones and PowerPC. What is the right name? The question is left as an exersice for the reader.

15.2. There is "snow" on my X desktop

How can I configure X so it removes the "snow" on my desktop?

The easy answer is: You can't. The kernel frame buffer device made by David Monro is still in an early stage, though working very well. Distortions in the picture when moving the mouse or scrolling a window are perfectly normal at eg 1024x768@60Hz. If you are a hacker, please fix it and post a patch to David. We would all love it very much.

15.3. I can't get my hardware to work

How can I get my new ultra whizbang XYZ card to work?

The 7248 is a PC−like box with ISA and PCI interfaces, so one should think that using "normal" PC hardware made for the x86 platform should work flawlessly. Sadly to say, it doesn't. The drivers often have to be ported, and there are not that many Carolina motherboard kernel hackers out there. In addition, much hardware made for the x86 platforms uses BIOS calls to work properly. As the 7248 and its relatives does not have such a BIOS, it's extremely difficult to get this hardware to run under Linux. Note: Linux−2.4 has fixed many of these problems. See below.

That said, there are working hardware for this box that runs with Linux. For questions about this, please contact the Workstation list, see Section 13.

Update: With the latest versions of the Linux 2.4 bk development tree (NOT the official Linux 2.4 sources), many of the problems stated above are fixed, and much more hardware is supported. For example did I put a standard eepro100 card in my box, and it worked flawlessly. This means you can use the 7248 for example as a packet−filtering firewall. I've also heard rumours on plain standard ISA Soundblaster cards working. Try and see if your card works. If it's interesting, send me an email, and I'll put a note here. See Section 12 for notes on building and installing a 2.4 kernel.

15.4. Me wants another distro!

Why do you not support more distributions in your howto?

There is one easy answer for this: I have not got the time to do this. I got LinuxPPC−1999 to work on my 7248 a year or so ago, and have not changed away from LinuxPPC since then. I plan to install Debian some time in the future, and have tried both SuSE and Yellow Dog, and have found that much of the steps are quite equal. The most troublesome part is to get the machine to boot the installation program. From there, each installer should do the job more or less itself. Getting X and sound to work should be more or less similar for all distributions, so just read the steps in this howto and try to make it work on your own. Feedback for this document is very welcome.

15.5. Boot floppies on other distributions

Can I use the floppies mentioned in this howto for other distributions

The bootfloppy mentioned in this howto should have support for all the hardware in the 7248, so you should be able to boot any distribution which uses the same floppy based installation scheme, that is, with the installation program on a ramdisk on one or more floppy disks. This means you should be able to boot LinuxPPC, SuSE, Yellow Dog and Debian with the bootimage. You have to have each distributions own installer ramdisk floppies, though.

15.6. The PReP boot partition?

Where should I mount the PReP boot partition?

To be able to understand the answer for this question, it's important that the reader understands how the 7248 boots into Linux. This is a three step procedure. First, the Firmware (which behaves in the same way as a PC BIOS) looks for something to boot. Usually, it should check the floppy drive, the CD drive, and then the first SCSI disk. On the SCSI disk it will look for a special partition called a PReP boot partition. On this partition, it will read the first program it can find there. If this is a Linux kernel bootloader, it will read and run this, and then the bootloader boots Linux. From here, Linux is in charge.

Many have asked where they should mount the PReP boot partition (the type 41 partition). This is a common misunderstanding. The PReP boot partition, usually located on /dev/sda1, should NOT be mounted anywhere. The files on this partition, usually only a single Linux kernel with a static linked kernel loader, are only used by the firmware when booting. The operating system does not use these files after the kernel has booted, so there is no need for mounting that partition.

Some people mix the meaning of the /boot directory and the PReP boot partition. Both use to contain kernels, but their use are different. /boot is used for storing kernels for later use, and for bookholding system info. The /boot directory is NOT read by the Firmware at boot time, so changing the contents of this directory does not change the way the Firmware loads Linux.

To be able to load a new kernel, you have to replace the existing kernel on the PReP partition. This is done with the dd command, see Section 10 for details.

15.7. It won't boot at all. Could it be bad RAM?

The machine won't boot at all. I suspect the RAM could be the problem. What kind of RAM should I use for this box?

The 7248 and it's cousines with Carolina motherboard do use special RAM, more specifically, they use only parity RAM. David Monro states that is is possible to make them work with other types of RAM if you remove the cache. Look at Section 13 for details.

16. Appendix: Using cfdisk to partition your harddisk

16.1. Hardisk names

SCSI harddisks are named with sdx, where x is a hardisk letter. The disk with the lowest SCSI ID on the first controller will become sda, the next lower sdb, an so on.

If you have IDE harddisks, they are called hda, hdb, hdc, and so on instead, where hda is the master disk on the first controller, hdb is the slave disk on the first controller, hdc is the master disk on the second controller, and so on.

16.2. Harddisk partitions

GNU/Linux systems usually use a partition scheme where a harddisk can have up to four primary partitions. If you want more, you have to make one of these an extended partition where you can make several logical partitions. The partitions are named with the disk they belong to, and a number. The first primary partition on the first SCSI disk is therefore sda1, the second primary partition is sda2, and so on. The first and second logical partition on an extended partition on the first SCSI disk is sda5 and sda6, and so on. If this makes absolutely no sense to you at all, try to read Section 17.

16.3. Starting cfdisk

you start cfdisk from the command line with the command

```
        cfdisk /dev/sdx
```

where x is the SCSI hardisk letter, like a, b, c, d, etc. So if I want to partition the first harddisk on the SCSI controller, I'll enter the command

```
        cfdisk /dev/sda
```

16.4. Using cfdisk

16.4.1. The user interface

After you have started cfdisk you'll get an interface where the current partition table is listed with the names and some data about each partition, and some command buttons on the bottom of the screen. To change between partitions, use the up and down arrow keys. To change between commands, use the left and right arrow keys.

16.4.2. Deleting a partition

To delete an existing partition, highlight it with the up and down keys, select the *Delete* command with the left and right arrow keys, and press Enter.

16.4.3. Making a new partition

To make a new partition, select the *New* command with the left and right arrow keys, and press enter. You'll get the choice between a primary and a logical partition. If you want a logical partition, the program will automatically make an extended partition for you. Then you must choose the size of the partition (in MB). If you can't enter a value in MB, return to the main screen with the Esc key, and select MB with the *Units* command.

16.4.4. Set the type of a partition

To set the type of a partition, for bootable PReP, Linux swap or Linux ext2, highlight the actual partition, and select the *Type* command. You'll get a list over different types. Press space, and you'll get even more. Find what type you need, and enter the number at the prompt.

16.4.5. Make a partition bootable

To be able to boot from a primary partition, you need to make it bootable. Highlight the actual partition and select the *Bootable* command.

16.4.6. Write the result to disk and quit

When you are content with the layout of the disk, select the *Write* command. The partition table will be written to disk. Remember that *this will destroy all data on partitions you have deleted* or changed. You should therefore be very sure that you want to do this before actually press the Return key.

To exit the program, select the *Quit* command.

17. Appendix: More on partitioning

After several questions on what partitioning really is, I'll just quote an answer I gave in a mail once.

Okay, here goes:

In an operating system you need several different filesystems for several different applications. For example, you need a swap filesystem because your main memory can't hold all information the operating system needs, so some of it has to be temporary written to disk. You may also need some special filesystem from which the machine reads the operating system when you switch it on. Finally, you need of course one or more filesystems to store the operating system program files and your user files. It may be a good idea to put these in different places (ie. on different filesystems) in case you have to reinstall the operating system, but don't want to scratch all your work.

The best thing is maybe to have all these filesystems on different disks. But one has seldom more than one or two disks in a computer. So what we do is to slice up the disk(s) in several slices (partitions) and use the slices for several filesystems. Then the operating system mounts the filesystems together to one single file tree, so it is easy to access the files.

(Other operating systems, like MS−DOS and NT use some other technology: They do not bind the slices together to one file tree, but keeps them separate as "stations". What is the best scheme? You figure!)

Here a thought example with one 2GB disk on a 7248: The mount point shows where in the file tree a filesystem is mounted.

```
    Partition   Size   Type             Mountpoint          Bootable
    −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
    /dev/sda1     10MB  41 (PReP Boot)     (Not mounted)    yes
    /dev/sda2    150MB  82 (Linux Swap)    (Not mounted)     −
    /dev/sda3   1840MB  83 (Linux ext2)  / (Root partition)  −
```

This would give a bootprompt command like this:

```
    root=/dev/sda3
```

If you want, you could add own partitions for important directories like /home, /boot, /var, /usr/local and so on. Here is an other example with two disks, actually my own configuration with two disks:

```
      Partition   Size   Type             Mountpoint          Bootable
      −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
      /dev/sda1     20MB  43 (PReP Boot)     (Not mounted)    yes
      /dev/sda2    133MB  82 (Linux Swap)    (Not mounted)     −
      /dev/sda5    930MB  83 (Linux ext2)  / (Root partition)  −
      /dev/sdb1    315MB  83 (Linux ext2)  /home               −
      /dev/sdb2    770MB  83 (Linux ext2)  /usr/local          −
```

This would give a bootprompt command like this:

```
      root=/dev/sda5
```

Before you ask:

- ext2 is Linux' standard filesystem

- Linux uses the old partition scheme from MSDOS. This means that if there are more than 3 partitions on one disk, one uses an extended partition (sda3) that may hold several logical partitions (sda5, sda6, sda7, ...)
- Yes, my partition scheme is a bad one. My root partition was filled up in a couple of weeks or so. Don't use it. It is an example only.

Hope this clears up some things.