Installing GNU/Linux on the IBM RS/6000 43P model 7248 HOWTO

Ingvar Hagelund

<ingvar@linpro.no>

Revision History

Revision 2.0	2003−06−02	Revised by: ih
2.0 is out.		
Revision 1.64	2003−06−01	Revised by: ih
YellowDog 3.0 (Sirius) is availble through dist−upgrade		
Revision 1.63	2003−05−26	Revised by: ih
Main parts of Suse, Yellowdog and Debian and Mandrake done.		
Revision 1.62	2003−02−26	Revised by: ih
Working on complete new version with several distributions.		
Revision 1.6	2002−12−19	Revised by: ih
Change from LinuxPPC to other distributions, first sketch		
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 several Linux distributions on the IBM RS/6000 43P model 7248 series.

Table of Contents

1. Introduction....1
1.1. Copyright Information and Legal stuff....1
1.2. Disclaimer and scope....1
1.3. New versions....2
1.4. Credits....3
1.5. Feedback....5
1.6. Translations....5

2. Overview....6
2.1. The IBM RS6000 43P 7248−132....6
2.2. What's the matter, why not use the original installation procedure?....6
2.3. What distributions are supported....7
2.4. An overview on what to do....7

3. Setting up the hardware with SMS....8
3.1. Where is the BIOS?....8
3.2. How to use the SMS....8
3.3. What settings to use....8
3.4. More info about the hardware....8

4. Get the installation files....9
4.1. Buying a CD....9
4.2. Download CD images over FTP or HTTP....9
4.3. Network installation....10

5. Make boot floppies....11
5.1. What floppies to make....11
5.2. How to make the boot floppies....11

6. Boot the machine and start the installation program....13
6.1. A note on partitions....13
6.2. Boot the machine and start the installer....13
6.3. The Debian installer....13
6.4. The SuSE installer....14
6.5. The Yellow Dog installer....14
6.6. The Mandrake installer....15
6.6.1. CD installation....15
6.6.2. Network installation....16
6.6.3. Installer notes....17

7. Post installation configuration....18
7.1. The Boot Prompt....18
7.2. Post installation configuration of Debian....18
7.3. Post configuration of SuSE....19
7.4. Post configuration of Yellow Dog....19
7.5. Post configuration of Mandrake....19
7.6. Installation: Done!....20

Table of Contents

8. Odds and ends....21
8.1. Network hangs....21

9. Compile a kernel....22
9.1. Why update the kernel....22
9.2. What files to download....22
9.3. Patching the source....23
9.4. Configuring the kernel....23
9.4.1. Initial kernel command string....23
9.4.2. Support for the integrated sound adapter....24
9.4.3. Support for framebuffer graphics....24
9.4.4. Support for other stuff....24
9.4.5. Save changes....24
9.5. Compiling and installing the kernel....24
9.6. Configure modules....25
9.7. Pray and reboot....25

10. Set up X....26

11. Resources....27
11.1. Other resources on Linux/PPC and 43P boxes....27
11.2. Installing other operating systems on the 7248....27

12. Todo....28

13. Frequently Asked Questions....29
13.1. XF68 or XF86....29
13.2. There is "snow" on my X desktop....29
13.3. I can't get my hardware to work....29
13.4. The PReP boot partition?....30
13.5. It won't boot at all. Could it be bad RAM?....30
13.6. Kernel boots, but stops at "Parity checking"....30

14. Appendix: Updating from YellowDog 2.3 (Dayton) to 3.0 (Sirius)....32

15. Appendix: Using cfdisk to partition your harddisk....35
15.1. Hard disk names....35
15.2. Harddisk partitions....35
15.3. Starting cfdisk....35
15.4. Using cfdisk....35
15.4.1. The user interface....35
15.4.2. Deleting a partition....36
15.4.3. Making a new partition....36
15.4.4. Set the type of a partition....36
15.4.5. Make a partition bootable....36
15.4.6. Write the result to disk and quit....36

Table of Contents

16. Appendix: More on partitioning.....37

17. Appendix: Make SMS and firmware floppies from Linux.....39
17.1. Why?.....39
17.2. How?.....39

1. Introduction

This document describes how to install GNU/Linux 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 workstation with a nice graphical user interfase. This relase covers several Linux distributions. Earlier, this document described the installation of several Linux distributions. Earlier, it only covered the LinuxPPC distribution, which is now obsolete. If you, for any particular reason, should be interested in versions of LinuxPPC, please have a look at my homepage .

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. Later I rewrote it in SGML and it is now a part of the LDP.

1.1. Copyright Information and Legal stuff

This document is copyrighted (C) 2003 Ingvar Hagelund and is distributed under the terms of the Linux Documentation Project (LDP) licence, stated below.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have any questions, please contact <linux−howto@metalab.unc.edu>

"Linux" is a registrated trademark owned by Linus Torvalds. "IBM" and "RS/6000" are trademarks owned by IBM Corporation. "MS−DOS" is a trademark owned by Microsoft Corporation.

Updated: June 1, 2003.

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 GNU/Linux 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 than described. Please don't ask me about this, as I have not tested others. Look in the Section 11 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 the 43P−133 are two terms for the same model, and the same goes for 7248−133 which I've also seen referenced from time to time. These three terms describes the exactly same model. 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 2.0

- 2.0 is finally out.
- Lots of updates. Now supports Debian, Mandrake, SuSE and YellowDog.

Version 1.64

- YellowDog 3.0 (Sirius) is availble through a dist−upgrade
- Heading for 2.0.

Version 1.63

- Most parts of Mandrake, YellowDog, SuSE and Debian are done
- Heading for 2.0.

Version 1.62

- Working on complete new version with several distributions
- Heading for 2.0.

Version 1.61

- Added an appendix on making SMS and firmware images from Linux

Version 1.60

- First sketch for other distributions

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

Installing GNU/Linux on the IBM RS/6000 43P model 7248 HOWTO

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

Nader Salasshahri

Thomas Junker

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 7248−132

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
- Carolina type motherboard
- 133MHz PowerPC model 604 processor
- 512K synchronous L2 cache
- Max 192MB RAM (?)
- 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 one 1.2GB IBM DPES 31080 SCSI hardisk

You can find more information at IBM's sales manual for this machine, found at http://www.ibmlink.ibm.com/. (click "United States", "SalesManual", "Continue", and search for the product number "7248" at the bottom)

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 Linux/PPC homepage/.

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

Most Linux distributions for PowerPC computers are aimed for the Apple PowerMac computer series. Those who support other hardware seems (naturally enough) to aim at quite modern hardware. The IBM 7248 series are not Macintosh computers nor very modern (although neither very old). This means that the standard kernels, the standard installation routines, and the standard system configurations do not fit our precious 7248 computers.

You should absolutely make no worries about this at all. 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. What distributions are supported

The term "supported" here is maybe too strong, as few distributions actually support the 7248. I will however in this document try to describe how to install the following Linux distributions on the 7248:

- Debian GNU/Linux 3.0 ("Woody")
- SuSE 7.3
- YellowDog Linux 2.3/3.0 ("Dayton"/"Sirius")
- Mandrake 9.1 ("Bamboo")

Some may ask where RedHat Linux is. RedHat does simply not support 32 bits PowerPC processors at all, so end of story. (Though there did exist a version of RedHat 7.2 for some special IBM PowerPC based hardware. It was never a success.) Those who are familiar with RedHat Linux should be quite content with YellowDog anyway, as it is based on RedHat, and feels very similar in use.

You should be able to build a Gentoo or Linux From Scrath installation on the 7248. See the respective Gentoo and LFS web sites for more information. You should probably be able to use some of the information given in this document, but I have personally never buildt a Gentoo or LFS installation, and probably never will, so no details on those will be given in this document.

2.4. 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
- Update the kernel
- Set up X

---

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 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 images for SMS and for updated firmware here. You should select MS−DOS images from the menu. They are actually zip−files in disguise, so you should be able to make those disks from a Linux box as well, using unzip and mtools. (If you feel totally lost on this, I wrote a quick help, look in Section 17)

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 sales manual, some clicking and searching from here.

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 one or several Linux distributions able to run on the 7248.

4.1. Buying a CD

- Debian GNU/Linux

  Buying Debian CDs for PPC might be a tricky one. You should scan this list for availble local (or shipping) resellers of Debian CDs, and ask them for a PPC distribution. You may be lucky.
- SuSE

  You should be able to find a local reseller, or buy directly off the net, try this link.
- YellowDog Linux

  You should be able to find a local reseller, or buy directly off the net, try this link
- Mandrake

  You should be able to find a local reseller, or buy directly off the net. Try this link: http://www.mandrakesoft.com

4.2. Download CD images over FTP or HTTP

The other way around is to download preformatted CD images, and burn them out yourself.

Before downloading installation images or doing a network installation, please consider buying boxed set, or services from the respective parties. Companies and organizations building Free Software needs support and money to survive.

- Debian GNU/Linux 3.0 "Woody"

  Find download sites for Debian GNU/Linux ISO images here. Please use a local mirror. Remember to pick the right platform, "powerpc".
- SuSE 7.3

  Download the SuSE installer cd here.
- YellowDog Linux 2.3 "Dayton"

  ISO images for YellowDog Linux 2.3 are availble for download from YellowDog's FTP site or from linuxiso.org. Only the first CD of the CD set is availble, but that's all you will need to install. Afterwards, you may want to upgrade the system over the net. You may also dist−upgrade to version 3.0 ("Sirius"). This is described in Section 14.
- Mandrake Linux 9.1 ("Bamboo")

  ISO images for Mandrake Linux 9.1 for PPC are availble for download at Mandrake's download page, please use a local mirror. You can also download from linuxiso.org

To make a CD, you need some CD burning software package. This is outside the scope of this document, allthough any decent CD burning software on any platform should be able to do it.

4.3. Network installation

All distributions mentioned in this document are able to do a network installation except for YellowDog. For the other distributions, both installation from CD and over the network will be described. The Mandrake installation is special, because it needs a local mirror, made from a complete ftp site mirror or from the CDs, see below.

All installation methods described in this document will be initiated with a floppy disk boot image. The 7248 is also capable of booting over the net (via tftp), and of course, by cdrom. This will not be covered in the following.

Before downloading installation images or doing a network installation, please consider buying boxed set, or services from the respective parties. Companies and organizations building Free Software needs support and money to survive.

5. Make boot floppies

Both installing via net or CD needs bootstrapping with floppy disks. In this chapter we will learn where to get floppy images, and how to make useable boot floppies from them.

5.1. What floppies to make

First we need a boot floppy. This will be a custom compiled Linux kernel image able to boot on the 7248. Then we need one or more ramdisk images.

- Debian Woody

  For Debian, we need a boot floppy and one ramdisk floppy. You can download necessary files from http://users.linpro.no/ingvar/43p/images/Debian/ . The files are called *debian−7248−boot.img* and *debian−7248−ramdisk.img* . (The ramdisk image is the root.bin from Debian boot−floppies)
- Yellow Dog Dayton

  For Yellow Dog we need a boot floppy, and no less than three ramdisk floppies, because of the size of the installer. I have built custom ramdisk images for the 7248. The files are called *ydl−7248−boot.img*, *ydl−7248−ramdisk−1.img*, *ydl−7248−ramdisk−2.img* and *ydl−7248−ramdisk−3.img*, and are available from http://users.linpro.no/ingvar/43p/images/YellowDog/. You will need all the ramdisk files.
- SuSE 7.3

  For SuSE, we need a boot floppy and one ramdisk floppy. You can download necessary files from http://users.linpro.no/ingvar/43p/images/SuSE/ . The files are *called suse−7248−boot.img* and *suse−7248−ramdisk.img*. (The latter is copied from ftp.suse.com)
- Mandrake Bamboo

  For Mandrake, we need a boot floppy and one ramdisk floppy. You can download necessary files from http://users.linpro.no/ingvar/43p/images/Mandrake/ . The files are *called mdk−7248−boot.img* and *mdk−7248−ramdisk.img*. (The latter is a copy of the all−r6sk.gz from the Mandrake cd)

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. If strange things things happen at boot time, try using another program for downloading the files, like wget or lynx.

5.2. How to make the boot floppies

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, so du a 'su root' before issuing the commands.

MS−DOS users may use the rawrite utility. You can download rawrite from several places, for example a RedHat mirror as ftp://ftp.uninett.no/. More information on how to use rawrite here.

To make the boot floppy, insert a floppy in the drive, cd to the directory containing the boot floppy image and

issue the following command, substitute "debian" to your distributon prefix if necessary.

```
dd if=debian-7248-boot.img of=/dev/fd0 bs=36b
```

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

To make a ramdisk floppy, insert a floppy in the drive, cd to the directory containing the ramdisk image, and issue the following command. Substitute the filename with an image for your distribution of choice, like "ydl−7248−ramdisk−2.img" for the second YellowDog ramdisk floppy image.

```
dd if=debian-7248-ramdisk.img of=/dev/fd0 bs=36b
```

Label the disk "Ramdisk floppy #1" or whatever you like. Remember that Debian and SuSE has one ramdisk floppy. For Yellow Dog, you need three.

---

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. A note on partitions

Below we will be asked about what partitions to make. We should have at least these partitions:

- A PReP 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.
- 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.

You may of course add as many other partitions to your system as you may wish, but this is the very minimum.

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.

For Debian and SuSE, we will use the *cfdisk* program for partitioning. For those unfamiliar with cfdisk and partitioning in general, I have made two small appendices on this, just because I am a very kind person. See Section 15.

6.2. Boot the machine and start the installer

To boot the 7248, just insert the boot floppy and turn the machine on. 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 floppies when prompted.

6.3. The Debian installer

You should just walk through the Debian installer like you would on any Debian installation. More information on how to use the installer can be found at your favourite Debian mirror, a good reference should be this document. You would maybe start here as you already have been walked through the preliminary steps. There are a few thing to remember though:

- Not all keyboard layouts are availble in the installer. Pick one you are able to use. We will have more to choose from once the system is installed.
- At the "Partition a Hard Disk" step, our partition program will be cfdisk. At this step we have to add the PReP partition as mentioned above, a swap partition and a root partition. Remember to write down on a piece of paper what partition to use for the root filesystem. Unless you have a really good memory, of course.
- The steps "Install Kernel and Driver Modules", "Configure Device Driver modules", "Configure PCMCIA Support", and "Install Foreign Modules" can be skipped, as all necessary driver are compiled into the boot floppy kernel.
- When installing the base system, we can choose either network or cdrom. Both should work. (I have even installed by floppy once, but this is something we really don't want to do.)
- Skip the steps "Make System bootable" and "Make a Boot Floppy" − they won't work. See the later chapter Section 9.5 on how to make the system boot from the harddisk.
- No bootloader will be installed, as there are no availble bootloader for the PReP PPC platform (except the one that's piggybacked onto the kernel). After the installation is done, you will therefore have to boot again from a floppy. See the next chapter for details.

When you are finished installing the base system, reinsert the boot floppy and choose "Reboot the system".

6.4. The SuSE installer

The SuSE installer should work without any hassle at all. Prelimenary, I've only done ftp install, but any installation method should do, provided you have access to the media. I have not been able to find an online version of the installation instructions, but it should be quite strightforward. If you have bought the boxed set, you have probably got a printed manual on dead trees. A few things to look up for anyhow:

- When partition the hard disks, remember to make a PReP boot partition as described above. As a general rule, use /dev/sda1 for this.
- If we're doing a ftp install, use the following path from a SuSE mirror root: suse/ppc/current/. An European mirror is available at *194.71.11.20* (ftp.sunet.se) in /pub/Linux/distributions/suse/suse/ppc/current/ . An American mirror is availble at *140.221.9.138* (mirror.mcs.anl.gov) in /pub/ftp.suse.com/ppc/current/ . There is a list of other mirrors here
- The download progress bar is broken on slow links, so we don't care about it unless we're connected to a fat pipe. Have patience.
- When the installation is finished, we may get a blank screen with nothing but a blinking cursor. Ensure that the installation is done (check for availble shells or status screens at VT2, 3 and 4. by pressing Alt+F2,F3,F4,F1, there should be none). Reinsert the boot floppy and power−cycle the box.

  No bootloader will be installed, as there are no availble bootloader for the PReP PPC platform (except the one that's piggybacked onto the kernel). After the installation is done, you will therefore have to boot again from a floppy. See the next chapter for details.

6.5. The Yellow Dog installer

Just before the installer starts, I have added a small pre−installation routine. Please follow the on−screen instrucions. For someone who has done some sysadmin earlier, the steps should be quite easy. For the beginner, it should not be to difficult. Hop to a virtual terminal by hitting Alt+F2. Type

```
cfdisk /dev/sda
```

to partition your first scsi harddisk. Change to sdb for your second, and so on. If you think this is a little difficult, I've written a small appendix on this Section 16. When done partitioning, run for example

```
mke2fs /dev/sda3
```

to make an ext2 filesystem on your third partition on your first hard disk. When you are finished, hop back to the main screen by hitting Alt+F1, and press enter to continue

You should walk through the installer as you would walk through any Yellow Dog installation (though only text−based interface is availble). I have not found any comprehensive online installation manual, but there is some info here. If you have bought a boxed set, you should open the box and Read The Fine Manual if you have questions not answered here.

There are a few things to look out for, though:

- At the "Installing Profile Selection" screen (one of the first screens), choose "Custom".
- At the "Installation Setup" screen, only "Local CD/DVD" is availble, so we'll have to choose ... well, you guessed it.
- At the "Partition Disks" screen, we "Edit" each disk, but just choose "Save" in the subscreens. "Add" and "Delete" won't work. And besides, we already have done this, haven't we?
- At the "Package Selection" screen, choose "Base Install", unless you have a full CD set. The downloadable CD image has only the base install packages.
- At the "Sound Setup" screen, we'll get an error message saying we are not one of those lucky bastards owning a Power Macintosh computer. As this is something we should be ashamed of. Ignore, sniff, and select "Ok". We'll fix sound later.
- At the "PReP Bootloader Installation" screen, the installer yells that it can't find a PReP partition. A little strange, as we just made one. (We did, remember to do that, didn't we?). This installer just can't get it, so continue to ignore it. Select "Ok".
- The "X11 Configuration" screen just flips by, so I guess that part was painless. Yeah, right.
- No bootloader will be installed, as there are no availble bootloader for the PReP PPC platform (except the one that's piggybacked onto the kernel). After the installation is done, you will therefore have to boot again from a floppy. See the next chapter for details.

6.6. The Mandrake installer

For the Mandrake installer, there are a few quite extensive prerequisites necessary. For a CD install, we need to replace the installer program on the first CD before burning out the CD ISO image. For a network install, we need a complete local ftp or http mirror, made from the Mandrake Bamboo ppc ftp directory, or the three CDs. Actually, we only need to replace one single file, but because the installer is unable to change package source during the installation (this is one reason why Debian is a wonderful distribution), we need local copies of all the files.

If we don't like to hazzle around and tune things, we'll stick to the CD installation. It's the easiest.

6.6.1. CD installation

First we have to download the three Mandrake ISO images from your favourite mirror. There exist ppc ISO images at least at a Sweedish mirror at ftp://ftp.chello.se/pub/linux/Mandrake−iso/ppc/.

Then we must change the first image by replacing the installer inside it. This could be done on any operating system able to mount a CD ISO image. The instructions below are made for running on RedHat Linux. We

presume the images are all put in /var/tmp.

First we mount the image by the loopback interface

```
mount -o loop=/dev/loop0 /var/tmp/MandrakeLinux-9.1-CD1.ppc.iso /mnt/cdrom
```

Then copy all files in the image to somewhere with plenty of space, eg. /var/tmp, and unmount the image again:

```
mkdir /var/tmp/bamboo1
cp -va /mnt/cdrom/* /var/tmp/bamboo1
umount /mnt/cdrom
```

Now replace the installer image:

```
cd /var/tmp/bamboo1/Mandrake/base
rm mdkinst_stage2.bz2
wget http://users.linpro.no/ingvar/43p/images/Mandrake/mdkinst_stage2.bz2
```

Finally rebuild the installer image and, if you want, remove the local copy of the cd contents:

```
cd /var/tmp
mkisofs -r -o MandrakeLinux-9.1-CD1.ppc.iso bamboo1/
rm -rf bamboo1
```

There. You now have a set of three working ISO images for the 7248. Burn them out on CDs. Then insert the boot floppy into the 7248 and turn the computer on. Disk-Jockey the ramdisk floppy when prompted. Just do a normal CD installation. See the installer notes below for details.

6.6.2. Network installation

First you need a local http or ftp mirror with a patched installer. This means that you need access to another computer able to run a web or ftp server. Any Linux distribution can do. A modern Windows or UNIX server should also be able to do the job. The trick is to download all necessary files, and change the installer file Mandrake/base/mdkinst_stage2.bz with a patched one. Below the the steps to get this done with the Apache web server on a RedHat Linux installtion, and with the CD iso image files, is described. (A complete download of the ppc archive from a Mandrake ftp mirror should work allright too, but I prefer to have the iso images availble.)

Unless already done, install and start the Apache web server on the system. This is described in the RedHat documentation. Then download the three Mandrake CD images from your favourite mirror. There exist ppc iso images at least at a Sweedish mirror at ftp://ftp.chello.se/pub/linux/Mandrake-iso/ppc/.

```
cd /somwhere/with/plenty/space
wget ftp://your.favourite.mirror/path/to/MandrakeLinux-9.1-CD1.ppc.iso
wget ftp://your.favourite.mirror/path/to/MandrakeLinux-9.1-CD2.ppc.iso
wget ftp://your.favourite.mirror/path/to/MandrakeLinux-9.1-CD3.ppc.iso
```

Mount the images via the loopback interface:

```
mkdir /mnt/bamboo1 /mnt/bamboo2 /mnt/bamboo3 /var/www/html/bamboo
mount -o loop=/dev/loop1 MandrakeLinux-9.1-CD1.ppc.iso /mnt/bamboo1
mount -o loop=/dev/loop2 MandrakeLinux-9.1-CD2.ppc.iso /mnt/bamboo2
mount -o loop=/dev/loop3 MandrakeLinux-9.1-CD3.ppc.iso /mnt/bamboo3
cd -
```

Copy the base files, and symlink the package directories, to a directory availble for the web server:

```
for i in base mdkinst share; do
cp -va /mnt/bamboo1/Mandrake/$i /var/www/html/bamboo/Mandrake; done
```

```
ln -s /mnt/bamboo1/Mandrake/RPMS1 /var/www/html/bamboo/Mandrake
ln -s /mnt/bamboo2/Mandrake/RPMS2 /var/www/html/bamboo/Mandrake
ln -s /mnt/bamboo3/Mandrake/RPMS3 /var/www/html/bamboo/Mandrake
```

Finally add my patched version of the installer image.

```
cd /var/www/html/bamboo/Mandrake/base
mv mdkinst_stage2.bz2 mdkinst_stage2.orig.bz2
wget http://users.linpro.no/ingvar/43p/images/Mandrake/mdkinst_stage2.bz2
cd -
```

You should now have a working local http mirror of the Mandrake installation files. Now boot with the boot floppy, and insert the ramdisk floppy when prompted. From the installation menus, select network install, select http and then give the address of the server where you just downloaded the files. The http directory from the example above is "/bamboo"

6.6.3. Installer notes

You will get an error when probing for a network and scsi drivers. The driver module files don't exist. We don't worry, as all necessary drivers are compiled into the running kernel.

If all package files necessary are availble, the installer should run without further errors.

The installer won't recognize the sound card, so sound won't work. Don't worry about this. We'll fix it in a jiffy. Same goes for XFree86, so we won't automagically get a working graphical user interface either.

No bootloader will be installed, as there are no availble bootloader for the PReP PPC platform (except the one that's piggybacked onto the kernel). After the installation is done, you will therefore have to boot again from a floppy disk. See the next chapter for details.

7. Post installation configuration

In this chapter we will boot our fresh installed system for the first time, and learn how to finnish the post−installation configuration of the various systems.

7.1. The Boot Prompt

If everything worked, you should now be able to reboot you're system, and start your freshly installed Linux system 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/sda3
```

I use sda3 as sda3 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). Luckily we did this, of course.

Note that we have to use this routine every time we boot machine, until we make our own kernel that suits the installation we just finished. This is described in Section 9

The system should boot up and maybe even doing some post configuration, see below.

7.2. Post installation configuration of Debian

The post installation configuration of Debian is described in detail in the document ch−init−config−en.html at your favourite Debian mirror. If you have cleared all previous stages, you get 400 bonus points, and can skip directly to paragraph 8.3.

By some strange reason, the Debian installer doesn't set up the network according to the fixes in the installer. If you want to use apt over a network connection, you should jump to a virtual screen (Alt+F2), log in as root, and set up networking. This is done by editing the file /etc/networking/interfaces , but the syntax of that file is way out of scope for this document. More information should be found in the Debian documentationm. Use the command

```
man interfaces
```

to get the manual page. When you are done, run

```
ifup eth0
```

to take the link up. Then jump back to the installer screen by pressing Alt+F1.

7.3. Post configuration of SuSE

The post installation configuration of SuSE should be quite painless. Make up a root password and enter it twice. Done. The rest should run automagically. Nice, eh?

7.4. Post configuration of Yellow Dog

YellowDog Linux does not have any post installation issues at all, except the expected kernel errors caused by lacking module directories. What a cool operating system! Look in Section 9 for detailed instructions on how to compile and install a working kernel.

There is some info at this url if we want to dig into more configuration. But we want to read the following chapters first.

Before taking another step you should go to a silent chamber and think: "I've got Yellow Dog 2.3. YES! But should I be content with that, now when YellowDog 3.0 is out? Should I really?" If the answer to that question is "No" then skip to Section 14. If you pass over the start field in the movement, you'll receive $2000 and a hotel.

7.5. Post configuration of Mandrake

Not much to mention here. The system should work more or less out of the box. Wow! You will probably get some errors on lacking kernel files. You will learn how to compile and install a complete kernel in Section 9. You may want to (re)configure your network. There does exist a tool called "drakconnect" that should be able to do this, but I never got it to behave. Configuring the local network is easy though. Just fire up your favourite editor (at least vi is installed) and edit the files mentioned below. This example describes a static ip configuration. Generally, this is the "RedHat" way to do things, so examples and documentation should be easy to find.

/etc/sysconfig/network

```
NETWORKING=YES
HOSTNAME=barky
GATEWAY=192.168.0.1
```

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

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.0.255
IPADDR=192.168.0.5
NETMASK=255.255.255.0
ONBOOT=YES
```

/etc/resolv.conf

```
nameserver 192.168.0.2
```

For a DHCP configuration, change BOOTPROTO to "dhcp", and skip the BROADCAST, IPADDR, NETMASK, GATEWAY and nameserver options.

I'm no Mandrake Guru. I actually never use Mandrake, and fixed the installer just for the exercise. By some reason, eth0 doesn't get active at boot time with the configuration above, though it does after a 'ifdown eth0;

ifup eth0', so I just put that in my rc.local. Go figure.

7.6. Installation: Done!

After the post installation configuration, the system should boot up to a ready state, and greet you with a login prompt. Congratulations, you have installed GNU/Linux on your 7248. You are dismissed to have a beer or a cup of tea. Or even coffee. 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 Introduction to Linux – A Hands on Guide by Machtelt Garrels.

8. Odds and ends

8.1. Network hangs

The following only affects 2.2 kernels.

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/init.d/network (or similar), 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. I really hope that somebody have a better fix on this later. Thanks to Doc Shipley for this tip.

9. Compile a kernel

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.

9.1. Why update the kernel

The kernel we have used this far is a complete 2.4 (2.2 for YellowDog) 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, you have to boot from 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-devel
ncurses-devel
kernel-headers
```

(This list should probably be longer.)

Note that the names of this packages may change slightly among the distributions.

9.2. What files to download

We'll use the latest 2.4 kernel sources with a few patches, among those the IBM–E15 frame buffer patch from David Monro, which gives us a working frame buffer console able to run XFree86. 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/ . When downloading the kernel source and patches, place all files in /usr/src .

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.

Note: You may use the standard Linux kernel source from any ftp.kernel.org mirror. A modern kernel like 2.4.21 should work allright. I've found the devel tree more well functioning on the 7248, so I stick with it.

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 .
chmod -R u+w .
```

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

Then, get David Monro's IBM E15 frame buffer patch from Leigh Brown's site page at http://www.solinno.co.uk/7043-140/files/2.4.19-2/

9.3. Patching the source

The next step is to patch the source files you just downloaded. First add the IBM E15 patch:

```
cd /usr/src/linux
patch -p1 < ../030-e15fb.diff
```

If you have more patches, add them the same way. Now the kernel source is ready for configuring.

9.4. 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 top directory of the kernel tree , /usr/src/linux .

In the top directory (/usr/src/linux) start the configuration program by issuing the following commands:

```
make ibmchrp_config
cp ingvar.config-2.4 .config
make menuconfig
```

Inside the kernel configuration system, we check over the different menus to get to know our possibilities. Don't be afraid to change anything. We can copy the original back, and load the configuration system again. There are a few things to check out before we go on compiling and installing:

9.4.1. Initial kernel command string

A thing you really should check, and possibly change, is the *Initial kernel command string*, located in the *General setup* submenu. This string is a space-separated list of options which are sent to the kernel at boot time. This is actually configuring the bootloader, and the closest we come to LILO, Grub or Yaboot on the 7248. Change the value of the root device to the device where your root (/) filesystem is mounted. This is done with *root=device*. An example could be *root=/dev/sda3*. If unsure, log in on another terminal and check with the *mount* command.

9.4.2. Support for the integrated sound adapter

The 7248 is equipped with a built−in Crystal Audio cs4232 sound adapter. There is support for this adapter in the Linux kernel. When configuring the kernel, check that the settings for sound are correct. They should look like this:

- Sound: *Y*
- OSS sound modules: *Y*
- Support for Crystal CS4232 based (PnP) cards: *M*

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

9.4.3. Support for framebuffer graphics

Some minutes ago we patched in a driver for the IBM E15 framebuffer driver. This is needed to run X (and to get a nice, smiling Tux when booting). If you for some reason don't want (or need) to run X, or you don't prefer a framebuffer console (it scrolls slower than standard VGA), remove support for the IBM E15 frame buffer in the *Console drivers* submenu.

9.4.4. Support for other stuff

Also check your own preferences on filesystems and other stuff you might have special need for or interest in. The downloaded config file should have sane values, though.

9.4.5. Save changes

When we have finished configuring, save our new configuration for later use, (at the bottom in the main menu) and select exit. Answer *Yes* to the question about saving the changes. You are now ready to compile the kernel.

9.5. 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:

```
cd /usr/src/linux
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 most Linux−based systems like, we keep kernels and their setupfiles in the /boot directory. Check what version you run with a look at the Makefile, and add proper version numbers in the commands below:

```
head /usr/src/linux/Makefile
cp /usr/src/linux/arch/ppc/boot/images/zImage.prep /boot/vmlinuz-2.4.21
cp /usr/src/linux/System.map /boot/System.map-2.2.18
```

Assuming that sda1 is your boot device with a PReP Boot partition, and your kernel has version 2.4.21, install the new kernel by issuing this command:

```
dd if=/boot/zimage-2.4.21 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-2.4.21 /boot/System.map
```

9.6. Configure modules

Before rebooting we must send some parameters to configure the sound modules. On YellowDog, SuSE and Mandrake, this is done in */etc/modules.conf*. In Debian, use */etc/modutils/sound*. Edit this file, and add the following lines:

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

On Debian, run */sbin/update-modules* when done.

9.7. Pray and reboot

There! We are ready to reboot. Light some candles (some people prefers to sacrifice chickens or even goats), remove the boot floppy, and issue the command:

```
reboot
```

If everything worked, 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. 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
- XFree86−100dpi−fonts and/or XFree86−75dpi−fonts
- gdm, kdm or xdm
- Maybe more packages here...

Note that these package names are for rpm−based systems. For Debian, do an

```
apt-get install xserver-xfree86 xfonts-100dpi xfonts-base xbase-clients xdm
```

to get the most important files.

Download my XF86Config−4.3, and copy it to /etc/X11/XF86Config. You can now start X with the command startx. On Debian you may start a graphical login screen with

```
/etc/init.d/xdm start
```

On YellowDog, SuSE and Mandrake, just hop to runlevel 5, and it will start a display manager for you.

```
/sbin/init 5
```

11. Resources

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

11.1. 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!)
- Linux on the RS/6000 7043−140 another old but still popular IBM PReP Workstation. Page by Leigh Brown. This page also has E15 frame buffer patches for Linux−2.4.
- 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 mailing lists. Especially interesting is of course the workstation list.
- PowerPC Linux project pages at sourceforge.net
- IBM's Linux on PPC project pages
- More info about Linux on PReP machines by Kazunori Aoshima
- IBM Sales Manual for the 7248−133 (Detailed description)
- PReP specification documents from IBM
- CHRP specification documents from IBM
- Cross Development for Linux/PPC from i386 by Matt Porter

11.2. Installing other operating systems on the 7248

- Of course, AIX should work like a dream...
- Rumors have been spread that once up in a time, even Window NT and Sun Solaris could run on this machine. Oh, well. Don't.
- NetBSD does of course work on PReP PowerPCs. It probably also works on your Remington typewriter from the early 50s. There is a a special page for our PReP architecture where the 7248 is mentioned explicit.

12. 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 in 2.2 kernels. If you know something about this, please let me know. Another solution is of course to stop using 2.2.
- Dual Boot with AIX. We need some kind of a bootloader. Does such a thing exist? Rumors says that Leigh Brown has something in the brewing.
- Get midi to work. Is this possible?

---

13. 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.

13.1. 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 mentioned in older versions of this document. 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 exercise for the reader.

13.2. 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 or Leigh. We would all love it very much. note that lower resolutions like 800x600 og even 640x480 works great. And no, there are only 8bit colors availble.

13.3. 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 always do. 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.

That said, there are working hardware for this box that runs with Linux. For questions about this, please contact the Workstation list, see Section 11.

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 9 for notes on building and installing a 2.4 kernel.

13.4. 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 bootloader, 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 9 for details.

13.5. 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. The spesifications are as follows: 72−pin SIMM, 5 Volt, Fast Page Memory with Parity, 70 ns. David Monro states that is is possible to make Carolinas work with other types of RAM if you remove the cache. Look at Section 11 for details.

13.6. Kernel boots, but stops at "Parity checking"

And now I thought it would work, but it stops at "Parity checking". I can't get a step further. Can you help me, please? Has this something to do with bad RAM chips? Or is it something wrong with my scsi devices?

You use a 2.2 kernel, don't you?

This message comes from the SCSI subsystem, so it has nothing to do with your RAM. Sometimes, by uknown reason, the Linux NCR driver in the 2.2−series caused the scsi controller to hang in some uninterruptible state, which endured, even bypassing reboot. The solution then was to boot AIX or even Windows NT for PPC (yes, such a beast exists, but you really don't want it), which resat the controller in proper condition. Alternatively, switch off the machine, pull out the battery inside, let it stay out for a couple of weeks or so, and fit things back together. The 2.4 driver fixed this problem.

Boot a 2.4 kernel, and you should be allright.

This could of course also be a real SCSI parity problem. If a 2.4 kernel doesn't help, check your SCSI devices for wireing and termination problems.

---

14. Appendix: Updating from YellowDog 2.3 (Dayton) to 3.0 (Sirius)

Here's how to update Yellow Dog Linux from 2.3 (Dayton) to 3.0 (Sirius). This assumes being done just after finishing the base install (ie: very few packages installed)

If we are running X (the following is based on the "base" install, so we probably aren't), switch to text mode (/sbin/init 3). We assume we have some network access, the fatter the better. We are going to download some packages manually, so check that you're able to do that. Some console−based download tool like ncftp (ftp) or lynx (http) might come handy.

Note for the following: If you get errors from rpm that are not noted here, you can always do an

```
rm -f /var/lib/rpm/__* ; rpm --rebuilddb
```

This repairs the rpm database in most cases.

First, ensure that we are running latest versions of all software. This may not be necessary, but whatever.

```
apt-get update; apt-get dist-upgrade
```

Now we change /etc/apt/sources.list to include repositories for 3.0 Note that there are only three repositories for 3.0: base, main, update. Fire up your favourite editor (at least vi should be installed) and update the file. A working sources.list may look like like this:

```
rpm ftp://ftp.uninett.no/linux/yellowdog/apt 3.0 base main update
rpm-src ftp://ftp.uninett.no/linux/yellowdog/apt 3.0 base main update
```

Now let's update the package list and check how far−fetched a complete dist−upgrade is:

```
apt-get update
apt-get -s dist-upgrade
```

Woha! Lots of errors there. Let's resolve the worst of them. Start with removing lots of packages. Note that after this, we can't read man pages.

```
rpm -e kudzu kernel-pcmcia-cs rpm-python yup yi pspell aspell groff man
```

What's status now?

```
apt-get -s dist-upgrade again
```

Still errors, and quite unresolvable. We have to take some more low−level action. The really pain is to get a new version of rpm installed, so let's start with that. Manually download the following packages:

```
  glibc-2.3.1-51a.ppc.rpm
  glibc-common-2.3.1-51a.ppc.rpm
  libelf-0.8.2-2.ppc.rpm
  popt-1.7-9d.ppc.rpm
  rpm-4.1-9d.ppc.rpm
  librpm404-4.0.4-8x.27.ppc.rpm
  apt-0.5.5cnc4.1-1b.ppc.rpm
  libgcc-3.2.2-2a.ppc.rpm
  libstdc++-3.2.2-2a.ppc.rpm
  expat-1.95.4-1.ppc.rpm
  fontconfig-2.1-3.ppc.rpm
  freetype-2.1.3-4.ppc.rpm
  XFree86-libs-4.3.0-2.1c.ppc.rpm
  XFree86-libs-data-4.3.0-2.1c.ppc.rpm
  XFree86-Mesa-libGL-4.3.0-2.1c.ppc.rpm
```

We don't have use for apt for a while, so let's just remove it:

```
rpm -e apt
```

So far, so good. Now, let's upgrade glibc and rpm. This is the most critical part. If we succeed in this, the rest is simple. If we get this wrong, a reinstall may be the only solution. rpm has some dependencies, so we have to include them now.

```
rpm -Uvh glibc-common-2.3.1-51a.ppc.rpm \
         glibc-2.3.1-51a.ppc.rpm        \
         libelf-0.8.2-2.ppc.rpm         \
         popt-1.7-9d.ppc.rpm            \
         rpm-4.1-9d.ppc.rpm             \
         librpm404-4.0.4-8x.27.ppc.rpm
```

If your setup is like mine, we're stuck with the following unresolvable error:

```
rpmlib(PartialHardlinkSets) <= 4.0.4-1 is needed by glibc-common-2.3.1-51a
```

This seems to be a problem overseen by the packagers of YellowDog (and to be honest, I think they inherited this error from RedHat), so we just ignore it, use some force, and hope this won't bite us later.

```
rpm -Uvh --nodeps --force glibc-2.3.1-51a.ppc.rpm                \
                          glibc-common-2.3.1-51a.ppc.rpm         \
                          libelf-0.8.2-2.ppc.rpm                 \
                          popt-1.7-9d.ppc.rpm                    \
                          rpm-4.1-9d.ppc.rpm                     \
                          librpm404-4.0.4-8x.27.ppc.rpm
```

Uh-oh. Ugly errors from rpm. Let's check if the rpm database still works.

```
rpm -qa
```

Well, looks like we've really done it. Now what? Sorry to say, but I've tricked you into ruining your computer installation. Get out and get a life instead of looking on these old dust-collectors. Just kidding. Luckily, rpm is able to rebuild it's databases, so we just remove the old one and build a new.

```
rm -f /var/lib/rpm/__*
rpm --rebuilddb
```

Now the database should work again, right?

```
rpm -qa
```

If this works, we collect 250 bonus points and two silver stars, and can continue our quest for 3.0. Find the dagger and throw it at the dragon.

If we don't have any special parameters for nsswitch and gconv-modules, we'll use the new ones (if existing):

```
mv -f /etc/nsswitch.conf.rpmnew /etc/nsswitch.conf
mv -f /usr/lib/gconv/gconv-modules.rpmnew /usr/lib/gconv/gconv-modules
```

Now let's reinstall apt. Is also have a lot of dependencies that have to be fixed:

```
rpm -Uvh apt-0.5.5cnc4.1-1b.ppc.rpm            \
         expat-1.95.4-1.ppc.rpm                \
         freetype-2.1.3-4.ppc.rpm              \
         fontconfig-2.1-3.ppc.rpm              \
         libgcc-3.2.2-2a.ppc.rpm               \
         libstdc++-3.2.2-2a.ppc.rpm            \
         XFree86-libs-4.3.0-2.1c.ppc.rpm       \
         XFree86-libs-data-4.3.0-2.1c.ppc.rpm  \
         XFree86-Mesa-libGL-4.3.0-2.1c.ppc.rpm \
```

I've sometimes got rpm to hang at this place. It's a known bug in rpm and can hit you anytime. If you experience this, get another terminal (Alt+F2 if you're on the main console) , kill rpm, and remove and rebuild the database. Then do the wanted rpm command again.

```
killall -9 rpm
rm -f /var/lib/rpm/__*
```

```
rpm --rebuilddb
```

Now; the installation of apt may have changed our sources.list, so check it out, and copy back the original if you want to.

```
mv /etc/apt/sources.list /etc/apt/sources.list.3.0.orig
mv -f /etc/apt/sources.list.rpmsave /etc/apt/sources.list
```

We should now probably resync the local apt database and check the status

```
apt-get update && apt-get -s dist-upgrade
```

Still some unresolved dependencies, but we're getting there. With apt installed, the rest is like chewing boiled fish. Nothing to care about. Ouch! a bone.

```
apt-get install pam
```

Drags in the following packages: chkconfig cyrus−sasl cyrus−sasl−gssapi cyrus−sasl−md5 cyrus−sasl−plain db4 hesiod openldap openssh openssh−clients openssh−server openssl perl perl−Filter python sendmail

Unless you have done some special pam configuration, use the new values for system authetication. Then go on updating.

```
mv -f /etc/pam.d/system-auth.rpmnew /etc/pam.d/system-auth

apt-get install passwd
```

Drags in the following packages: glib2 libuser

```
apt-get install kbd
```

Drags in the following packages: initscripts util−linux

```
apt-get install compat-db compat-pwdb
```

Drags in the following packages: compat−libstdc++

```
apt-get -s dist-upgrade
```

If no errors, we're with all the difficult parts. Good! Now the final hurdle: A full dist−upgrade. Do an

```
apt-get -y dist-upgrade
```

and enjoy getting to the end of this quest. You get all the points. Just kiss the frog and you'll get the princess too. You may want to reboot, just to check that everything comes up like you wish.

You may want to reinstall some packages again. For example "man" may become handy.

```
apt-get install kudzu rpm-python pspell aspell groff man
```

That's all there is to it. It wasn't that difficult, was it?

---

15. Appendix: Using cfdisk to partition your harddisk

15.1. Hard disk 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.

15.2. Harddisk partitions

GNU/Linux systems often uses a partition scheme inherited from MS−DOS. With this, 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 16.

15.3. 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
```

15.4. Using cfdisk

15.4.1. 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.

15.4.2. 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.

15.4.3. 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.

15.4.4. 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.

15.4.5. 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.

15.4.6. 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.

16. 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
- GNU/Linux often uses the old partition scheme from MS−DOS. This means that if there are more than 4 partitions on one disk, one uses an extended partition (sda4) 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.

---

17. Appendix: Make SMS and firmware floppies from Linux

This appendix will show how to make SMS and firmware floppies from Linux or another UNIX−like operating system

17.1. Why?

Some people, like myself, seldom have access to boxes running AIX or MS−DOS, so I thought it could be handy to show how to make these utility floppies from a system running Linux (or any other UNIX like box).

17.2. How?

To do this, you need the mtools package and the unzip program. Any reasonable Linux distribution should include these tools. I also use wget for downloading.

You will have to download the last version of SMS and firmware from IBM. If you want more information on the files, look here.

You also have to get a password to access the files. To obtain this, you have to agree to this EULA.

The downloaded file is nothing but a ZIP−archive in disguise. Download it to a suitable directory, and unzip ip like this:

```
mkdir ibmdownload; cd ibmdownload
wget http://techsupport.services.ibm.com/server/mdownload2/7248100.exe
unzip 7248100.exe
```

You will have to enter the mentioned password when prompted.

In the contents of the archive, there's among the files another ZIP archive, containg the SMS software. Put the contents of this into a subdirectory like this:

```
mkdir sms; cd sms
unzip ../sms107.exe
```

Now label a floppy disk "SMS", and put it into your floppy drive. Then format the floppy and copy the SMS files to it like this:

```
mformat a:
mcopy *.* a:
cd ..
```

Remove the floppy, insert another labeled "Firmware", and continue with the firmware files:

```
mformat a:
mcopy *.6xe a:
mcopy p93h4940.im* a:
cd ..
```

Woho! You're done. Now that wasn't difficult, was it?