Andreas Schwab, Karl Eichwalder, Stefan Fent, Werner Fink, Peter Findeisen, Dennis Geider, Viviane Glanz, Carsten Groß, Roland Haidl, Björn Jacke, Richard Jelinek, Hubert Mantel, Johannes Meixner, Edith Parzefall, Peter Pöml, Stefan Reinauer, Peter Reinhart, Marc Rührschneck, Thomas Schraitle, Klaus G. Wagner, Christian Zoz

SuSE Linux 7.2

System and Reference Manual

SuSE Inc.
580 2nd Street, #210
Oakland, CA 94607
USA
Toll free phone number
within the US and Canada: 1-888-UR-LINUX (1-888-875-4689)
Phone.: +1-510-628-3380
Fax.: +1-510-628-3381
e-mail: info@suse.com
orders@suse.com
webmaster@suse.com
WWW: http://www.suse.com

Europe:

SuSE Linux Ltd.
The Kinetic Centre
Theobald Street
Borehamwood, WD6 4PJ UK
Phone.: +44-20-8387-4088
Fax: +44-20-8387-4010
WWW: http://www.suse.co.uk

SuSE GmbH
Schanzäckerstr. 10
D-90443 Nürnberg
Germany
Tel.: +49-911-740-5331
Fax.: +49-911-741 77 55
e-mail: suse@suse.de
WWW: http://www.suse.de

Andreas Schwab, Karl Eichwalder, Stefan Fent, Werner Fink, Peter Findeisen, Dennis Geider, Viviane Glanz, Carsten Groß, Roland Haidl, Björn Jacke, Richard Jelinek, Hubert Mantel, Johannes Meixner, Edith Parzefall, Peter Pöml, Stefan Reinauer, Peter Reinhart, Marc Rührschneck, Thomas Schraitle, Klaus G. Wagner, Christian Zoz

SuSE Linux 7.2
IA64
System and Reference Manual

1st edition 2001
SuSE GmbH

Layout: LaTeX
English Translation and Editing: Rebecca Ellis, Marc Rührschneck
Cover design based on graphics by Stephan Endraß.
Design by Manuela Piotrowski.
Geeko-Icons by Rolf Vogt.

Contents

1 Introduction

1.1 What Is “Linux”?

The actual “Linux” is the *kernel*, the “heart” of every UNIX operating system.

But the kernel alone does not constitute an entire operating system. Especially in UNIX there is a huge amount of free software available; this means that practically any standard tools in Unix are also available in Linux. It is these programs which make the operating system what it is.

Many of these tools are GNU versions (GNU) of these Unix programs and many of them offer enhanced features over the originals. Of these, the GNU C/C++ compiler, one of the best compilers available, is probably the most famous program. We should not forget the many tools, small and large, which can be used on the command line, in shell scripts or as fully fledged programs; from the shell, file and text utilities such as sed, awk and perl to editors like vi or complete work-environments like Emacs.

What makes Linux complete is XFree86™ (current version, 4.0.3), the X Window System (currently X11 Release 6.4) for PC-based UNIX systems. This port is part of the official X11R6.4 distribution of the X Consortium, Inc., and is therefore fully, compatible with that standard.

All these components, together with many other tools, applications and amusements (e. g., games), make up the system known as *Linux*. There is a large amount of free software available, and there are many ways to put a Linux system together.

At this point the Linux distributions come into play, in this case, SuSE Linux. A distribution is a selection from a wide palette of free software. The end product is, in the case of the SuSE Linux CDs, over 1000 software packages.

Note

SuSE Linux includes, in *addition* to free software – software in which the source-code is included on the CD-ROMs, software packages that, for different reasons, SuSE can only offer in compiled form.

Due to the fact that distributions are offered free on the Internet, updating your system does not necessarily mean buying a new distribution.

1.2 Purpose of this Book – Notes on Usage

This book should not and cannot be a replacement for more detailed literature which should allow an interested user to dig deeper into the fascinating world of *high-performance computing* – for this, refer to the bibliography at the end of the book (Chapter 13.6 page 344).

The dynamics of Linux and of free software is a big challenge as far as writing a handbook is concerned. It must keep up with the most recent developments and, at the same time, remain suited for those new to the Unix world without boring experienced Linux users. The book in front of you attempts to do justice to all these requirements.

The aim of this book is to allow Linux beginners an encouraging start on a voyage of discovery through the new system.

The book is basically divided into the following sections:

Installation: This section guides the Linux novice with a documented example installation (Chapter 2 page 9). It points out possible sources of error and provides concrete help with problems.

YaST 2: The usage of *YaST* 2 , the SuSE installation and administration tool, is described in Chapter **??** page ??.

YaST: The usage of *YaST* , a text based tool to configure SuSE is described in Chapter 3 page 51. Chapter 4 page 93 explains how to boot the kernel.

Graphical User Interface: Activating and setting up the X user interface is the subject of Chapter 5 page 125.

Sound, Scanner, Notebooks, Printing etc.: In Chapters 6 page 173 through 8 page 205 the capabilities of your hardware are explained.

The Linux Kernel: Here we get down to the heart of the matter. The Linux kernel is introduced in Chapter 9 page 237 and Chapter 10 page 245, providing you with a guide on how to compile and install your own kernel.

Update, Software Packages, Booting: Several update strategies as well as software management are introduced (Chapter 11 page 279), special features of SuSE Linux are covered and the boot concept is explained (Chapter 13 page 311).

Special Features of SuSE Linux: Here you will learn more about the file system and the TEX environment of SuSE Linux. Special software packages and system configuration files will also be discussed here.

The SuSE Linux Boot Concept: In this Chapter 13 page 311 you will read details on the init program and the several runlevels.

KDE2 - Usage and Configuration: Here you will learn how to use KDE2, the default graphical desktop for SuSE Linux. Konqueror, the file manager of KDE2 will also be introduced here.

Working With the Shell: In Chapter **??** page ?? you will get some useful information on the shell. So, you are able to use the system with a text based interface.

Help: Here we will give you ways to get help online.

Technical Appendix In the appendix you can find a list of the most important configuration files, the most frequently used key combinations, example files, a licensing agreement, details on the SuSE support, and much more.

Bibliography and Index If there is something you still haven't found yet, you should consult the extensive bibliography. Either to find the place which explains important concepts, or simply to increase your general knowledge about Linux, Unix or computers.

1.3 Typographical Conventions – or What Does “earth:˜ # **ls**” Mean

The typographical conventions used in this guide are explained in Table 1.1.

Text layout	Meaning
LINUS TORVALDS	important persons
GNU Emacs (**emacs**)	the program GNU Emacs, is invoked with the command **emacs**
Adabas D	the product Adabas D
`/etc/passwd`	file or directory name
<file>	when entering a command, `file` should be replaced by the actual value; you should *not* enter the angled brackets!
PATH	the environment variable with the name PATH
`192.168.1.2`	value of a variable
ls	the command **ls**
‘news’	the user ‘news’
earth:˜ # **susehelp**	‘root’-shell in the directory ~ (‘~’ stands for the “home directory” of the respective user), you should enter the command **susehelp** – here earth is an example of the name of a computer (hostname).
newbie@earth:/tmp > **ls**	in the shell of the user ‘newbie’ in directory /tmp, type the command **ls**
C:\> **fdisk**	at the DOS prompt, type the command **fdisk**
Alt	key to press, here the key “Alt”
Ctrl + Alt + Del	when ‘+’ is “added” all keys are to be pressed at the same time; keys to be pressed sequentially are separated by a space only
"Permission denied"	system messages
‘System update’	menu entry ‘System update’
Duesentrieb	company “Duesentrieb”

Table 1.1: Typographical conventions used in the text layout

1.4 Additional Linux Documentation

Support Database

Many solutions to well-known problems are already available. These are presented in our Support Database at http://sdb.suse.de/en/sdb/html/index.

html. We recommend that you look there first!

Besides the online Support Database, the complete Support Database can also be found in package sdb_en, series doc on your SuSE Linux CD. Those articles can be read using any WWW browser.

Hypertext Help

A major part of the documentation is available in *hypertext* form. To start the hypertext system, run **susehelp**. Depending on whether X Windows is running, a browser with or without graphical interface will be started for reading the documentation.

Additional options for the help system can be displayed by invoking

```
newbie@earth:/home/newbie > susehelp --help
```

Texinfo and Info

Some program packages include documentation in Texinfo format, which is another hypertext variant. These files can be read with Infoviewer (**info**) or by using the Emacs (**emacs**) info mode. In X Windows, you can use tkInfo (**tkinfo**) or the older xinfo; on the other hand, XInfo (**xinfo**) is only a bare-bones X program. In addition, these documents can be read in the SuSE help system.

Manual Pages

The usual way to get information about programs or commands is to invoke the command **man**.

```
newbie@earth:/home/newbie > man <command>
```

displays an overview of the tasks and options of each command.

The manual pages – often abbreviated to “manpages” – can be read via the SuSE help system; see section 1.4.

FAQ’s, HOWTO’s and README’s

The directory /usr/share/doc contains subdirectories with information about the corresponding packages. There you will often find the missing command option, the name of the configuration file you could not find anywhere else, etc. In any case, it is worth looking there before deleting the software. The directory /usr/share/doc/faq should be pointed out. In it, lists of frequently asked questions and appropriate answers on a wide variety of problems are placed if the package manyfaqs, series doc is installed. The directory /usr/share/doc/howto contains “recipes” explaining how to install certain packages or what to do when you encounter problems.

In Linux the program less is used to read files. Although the file is compressed (.gz), less is smart enough to handle this.

```
newbie@earth:/home/newbie > cd /usr/share/doc/howto
newbie@earth:/usr/share/doc/howto > less DOS-to-Linux-HOWTO.gz
```

In the directory `docu/howto` on the first CD there can be found the most current versions, at the time of the CD going to press, of the *most important* HOWTO files. It is certainly worth taking a look in this directory! These files are uncompressed; they can therefore be read comfortably, even before installation (in DOS, for example, with **`type`**).

Kernel-specific questions and answers can be obtained from `/usr/src/linux/Documentation`. This is only available if you have installed the kernel source (package `linux` or package `lx_suse`), which is highly recommended anyway. Furthermore, there are many useful hints in the kernel source subdirectories, and, for the brave, even the kernel sources themselves.

If you have a question which is not answered in this book, please look at the sources for more information since the scope of this book is limited.

Free Books

The package `books`, series `doc1` (Documentation) contains some books in PostScript format. You can view these books with the package `gsview`, series `gra1` (Graphics) or package `gv`, series `gra1` (Graphics) – if you don't care about trees you can print them as well. Before printing, you should ask yourself whether it might be more economical to buy the book.

1.5 Acknowledgments

Besides everyone who has contributed to the large success of Linux, we would like to thank Eberhard Moenkeberg, who was very helpful with CD-ROM driver problems.

Many thanks go to Hans Lermen, author of the well-known loadlin.exe, which allows Linux to be started from DOS and the DOS setup.exe of this Linux system.

Thanks also to all those who sacrificed their spare time for making Linux available to a vast number of international users by helping translating YaST and linuxrc texts: Jesús Bravo Álvarez, Gunay Arslan, Jaume Badiella i Aguilera, Zbigniew Baniewski, Sándor Bárány, Olaf Borkner-Delcarlo, Michael Bravo, Görkem Çetin, Franca Delcarlo, Jochen Depner, Benedek Hermann, Ibán José García Castillo, Dora Georgarou (with Romy the barking dog), Jonas-Gintaras Gocentas, Pablo Iranzo Gómez, Krzysztof Hotiuk, Milan Hromada, Arthur Kerkmeester, Ralf A. Lanz, Françoise Lermen, Zoltán Levárdy, Nuno Lima, Francisco Javier Moreno Sigüenza, Tomas Morkus, Karine Nguyen, Matts Nordman, Aleksey Novodvorsky, Razvan Cristian Oprea, Gerco Oudhof, Jordi Jaen Pallares, George Papamichelakis, Alexey Pavlov, Ákos Rátkai, Bruno David Rodrigues, Voula Sanida, Gabor Sari, Aleksey Smirnov, Laszlo Varadi, Steve Varadi, P. Vlachodimitropoulos, Joao Teles, Nuno Vieira, January Weiner, Monika Weiß and I Made Wiryana. The meticulous yet riveting task of coordinating this all was undertaken by Antje Faber.

Many thanks, of course, to the dauntless beta testers who risked their running systems: Andreas Koegel, Christian Hüttermann, Dirk Ulbrich, Eberhard Mönkeberg, Frank Hofmann, Georg C. F. Greve, Harald König, Harald Wieland, Karlo Gross, Jens Frank, Lutz Pressler, Martin Hehl, Martin Konold, Martin Schulze, Michael Kleinhenz, Norbert Eicker, Oliver Zendel, Ralf Geschke, Stefan Bliesener, Thomas Wörner, Ulrich Goebel, Ulrich Windl, Volker Lendecke, Wolfgang Barth, and the KDE Team.

Matthias Eckermann has re-worked the LATEXmacros and modernized the layout of the book, and Manuela Piotrowski provided graphics guidelines. These were based on the KOMA-""Documents class by Frank Neukam and Markus Kohm, which were in turn built around the “old” `susebuch` macros by Werner Fink and Martin Scherbaum.

The following were involved in the production of the previous manual texts: Bodo Bauer, Daniel Bischof, Michael Burghart, Roland Dyroff, Christian Egle, Klaus Franken, Jürgen Geck, Olaf Hering, Marc Heuse, Florian La Roche, Ralf Lanz, Volker Lendecke, Hans Lermen, Arndt Mehlhorn, Jay Migliaccio, Norbert Munkel, Jordi Jaen Pallares, Christoph-Erdmann Pfeiler, James Samuel, Martin Scherbaum, Burchard Steinbild, Jörg Strebel, Thorsten Wandersmann, Udo Weber and Stefan Wintermeyer.

Valuable references – and even partially for the text material:

Marco Banse, Stephan Barth, Martin Bergmann, Manfred Blumenthal, Bettina Botter, Stefan Dirsch, Uwe Drechsel, Thomas Fehr, Peter Findeisen, Matthias Frühauf, Kurt Garloff, Reinhard Gerndt, Lenz Grimmer, Iris Grossmann, Berthold Gunreben, Michael Hager, Torsten Hallmann, S. H. A. Handgraaf, Michael Hasenstein, Ulrich Hecht, Fabian Herschel, Bernhard Hölcker, Kevin Ivory,Andreas Jaeger, Klaus Kämpf, Karsten Keil, Andi Kleen, Sabine Kleine, Marcus Kraft, Thorsten Kukuk, Werner Kurk, Frank Lemser, Hans Lermen, Dirk Lerner, Christopher Mahmood, Walter Mecky, Marco Michna, Rüdiger Oertel, Ali Özcan, Norman Pancner, Dirk Pankonin, Petr Pavlík, James Ogley, Siegfried Olschner, Milisav Radmanic, David Rasch, Andrea Ratschker, Marion Reyzl, Rupert Rockinger, Anja Schambach, Manfred Schlenzka, Robert Schmid, Andreas Schmidt, Helmut Schoeberl, Dirk Spilker, Roger Whittaker, Sandra Wiegratz, Steffen Winterfeldt and Olaf Zaplinski. Production technical support took place thanks especially to Holger Dyroff, Christian Egle, Claudio Taulli and Nicole Zwerenz. Thanks and keep up the good work!

And the many important people without whom this would never have been possible: to all, a big, big THANK YOU!

Further thanks go to STEFAN ENDRASS, who computed the cover picture of the CD.

*

*   *

*

The global team of Linux developers is still working intensively on Linux – most of them on a voluntary basis. We want to thank them for their efforts – this CD distribution would not exist without them. Our work aims at making their work accessible to a large number of interested users.

Last but not least, special thanks to PATRICK VOLKERDING for supporting our work and, of course, many thanks to LINUS TORVALDS!

And last, but not least: Frank Zappa and Pawar!

Nuremberg, 18th June 2001

Have a lot of fun!

Your SuSE Team

2 Installing SuSE Linux

2.1 Installation with YaST2

2.1.1 Introduction

On the following pages, you will find instructions for installing SuSE Linux with YaST2. We will show you a quick and easy way to run Linux on your computer.

2.1.2 Starting Your System from CD–ROM

To prepare the drive, start your computer and insert the first SuSE Linux CD or DVD containing the necessary Linux configuration applications into your CD ROM.

SuSE Linux will now be started for installation from the selected medium.

Boot Settings

If the machine does not boot from the CD–ROM, it will have to be started with the SuSE Linux boot floppy. The start screen will appear and the installation will begin.

If your machine does not boot from the CD–ROM, you will have to change the settings in the computer's BIOS.

At initial start-up, a diagnosis of the computer's hardware will be carried out and, among other things, the random access memory will be tested. You will be able to recognize this at the system memory count. At the bottom of the screen, you will be told which key calls up the BIOS setup is. Usually, this is the `Del` or `F1` key. Press the key specified and the BIOS setup will appear.

If your machine is equipped with an AWARD BIOS, this entry will be labeled BIOS FEATURES SETUP. Other manufacturers use similar entries such as ADVANCED CMOS SETUP. Select the corresponding entry and confirm with ↵. Relevant for your purposes is the sub-item normally listed as Boot Sequence (of the drives). The default setting for this is usually C, A or A, C. In the first example, when booting, the machine will search the operating system on the (C) drive and then, on the (A) drive.

Select '`Boot Sequence`' and then press the `Bild ↑` and `Bild ↓` keys until the `A, CDROM, C` sequence is shown.

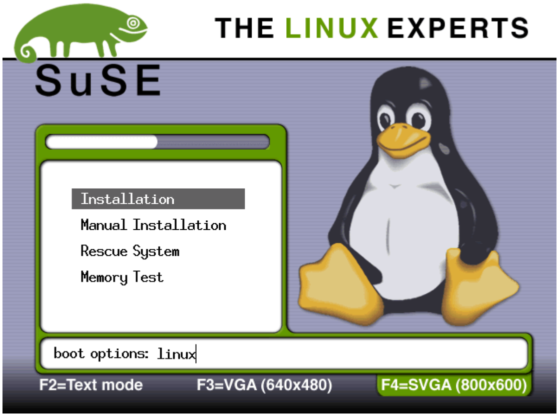
Next, exit these settings by pressing the `Esc` key. In order to save the changes, select '`SAVE & EXIT SETUP`' or press `F10`. At this point, you will be asked if you want to exit the BIOS setup and save these new settings.

If you have a SCSI CD-ROM drive, you will have to call up its BIOS – in the case of a Adaptec host adapter, with Strg + A. Select '`Disk Utilities`'; the system will check the hardware installed and display the corresponding information. Make note of the SCSI ID for your CD-ROM. Next, exit the menu with Esc in order to open '`Configure Adapter Settings`'. You will find '`Boot Device Options`' under '`Additional Options`'. Select this menu and press ↵. Now enter the CD–ROM drive ID you wrote down before and again, press ↵. You will return to the start screen in the SCSI BIOS by pressing Esc twice, which can be exited by confirming with '`Yes`' to reboot the machine.

2.1.3 The Opening Screen

A screen as in Figure 2.1 shows you that SuSE Linux is ready to be booted for the installation. Be sure to select '`Installation`' here (the default selection). Either wait a few seconds or just press Enter to load the kernel.

A few seconds later, SuSE Linux starts loading a minimal Linux system which takes over the rest of the installation procedure. A number of messages and copyright notices will then appear on the screen. At the end of the loading process, the YaST2 program will start, and a few seconds later, the graphical interface of YaST2, the SuSE Linux installation program, will be displayed.



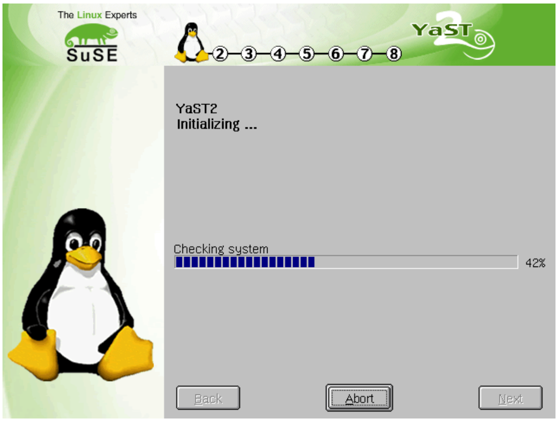
boot options: linux

Figure 2.1: The initial boot screen

2.1.4 YaST2 Takes Over

Now the actual installation of SuSE Linux starts with the YaST2 installation program. Figure 2.2 on the next page shows you what the screen will look like. During this phase, the hardware available on your system is checked and prepared for the installation. A bar in the middle of the screen shows the progress of the installation.

All YaST2 screens have a common format. On the left, help texts are shown, providing information on the current help topic. All entry fields, lists and buttons on the YaST2 screens can also be accessed by your mouse. If your cursor doesn't move, your mouse has not been automatically recognized by Linux. You will then need to use your keyboard.



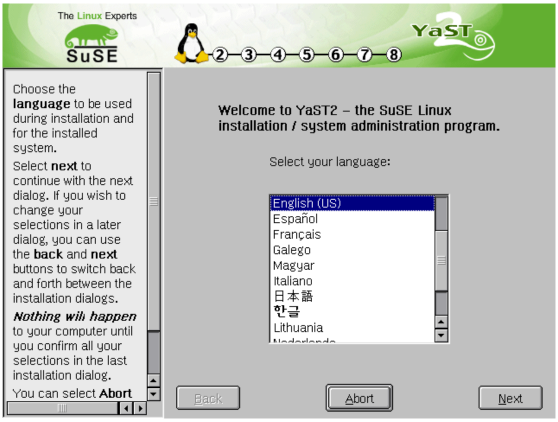
YaST2
Initializing ...

Checking system 42%

Figure 2.2: The hardware analysis

2.1.5 Selecting a Language

SuSE Linux and YaST2 are adapted to use the language you have selected. English is the default setting for the English distribution of SuSE Linux. These settings can be changed individually.



Choose the **language** to be used during installation and for the installed system.

Select **next** to continue with the next dialog. If you wish to change your selections in a later dialog, you can use the **back** and **next** buttons to switch back and forth between the installation dialogs.

Nothing will happen to your computer until you confirm all your selections in the last installation dialog.

You can select **Abort**

Welcome to YaST2 – the SuSE Linux installation / system administration program.

Select your language:

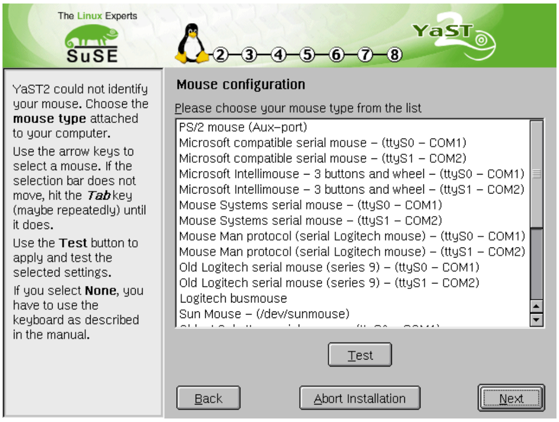
- English (US)
- Español
- Français
- Galego
- Magyar
- Italiano
- 日本語
- 한글
- Lithuania
- Nederlands

Figure 2.3: Selecting the language

If your mouse cursor still doesn't work, press the Tab key repeatedly until the '`Next`' button appears, then press the ↵ key.

2.1.6 Mouse Pointer

If YaST2 didn’t recognize your mouse type automatically, an entry screen will appear as shown in Figure 2.4.



YaST2 could not identify your mouse. Choose the **mouse type** attached to your computer.

Use the arrow keys to select a mouse. If the selection bar does not move, hit the ***Tab*** key (maybe repeatedly) until it does.

Use the **Test** button to apply and test the selected settings.

If you select **None**, you have to use the keyboard as described in the manual.

Mouse configuration

Please choose your mouse type from the list

```
PS/2 mouse (Aux–port)
Microsoft compatible serial mouse – (ttyS0 – COM1)
Microsoft compatible serial mouse – (ttyS1 – COM2)
Microsoft Intellimouse – 3 buttons and wheel – (ttyS0 – COM1)
Microsoft Intellimouse – 3 buttons and wheel – (ttyS1 – COM2)
Mouse Systems serial mouse – (ttyS0 – COM1)
Mouse Systems serial mouse – (ttyS1 – COM2)
Mouse Man protocol (serial Logitech mouse) – (ttyS0 – COM1)
Mouse Man protocol (serial Logitech mouse) – (ttyS1 – COM2)
Old Logitech serial mouse (series 9) – (ttyS0 – COM1)
Old Logitech serial mouse (series 9) – (ttyS1 – COM2)
Logitech busmouse
Sun Mouse – (/dev/sunmouse)
```

Figure 2.4: Selecting the mouse type

To select your mouse type, use the (↑) and (↓) keys. If you have documentation for your mouse, this should provide you with a description of the mouse type. Select the mouse type from the list. Confirm your selection either by

- pressing the (Alt) + (T) keys or
- pressing (Tab) and then confirming this with (↵).

Now you can test to see if your mouse is working. If the mouse cursor on the screen follows your movements, then your mouse is configured correctly. If the cursor doesn’t move, select a different mouse type and try again.

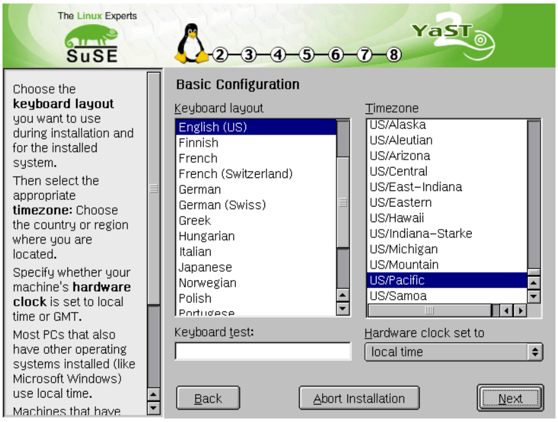
2.1.7 Keyboard and Time Zone

The next step (Figure 2.5 on the next page) involves the keyboard layout and the time zone. In the field ‘`Hardware clock set to`’, you can choose between `local time` and `GMT`. Your selection depends on the clock settings in the BIOS of your computer. If this is set to `GMT`, SuSE Linux will automatically apply the time change for Daylight Savings, Standard Time, and vice versa.

Now select the desired keyboard layout. Usually, this corresponds to the language you chose. Select the correct time zone in the other column.

2.1.8 Update or New Installation

Now you can choose either a ‘`New installation`’ or an ‘`Update`’ installation. The ‘`Update`’ installation is only for systems that already have a SuSE Linux version installed. Click on ‘`Next`’ to resume installation (Figure 2.6 on the facing page). In this chapter, we will only describe ‘`New installation`’.



Choose the **keyboard layout** you want to use during installation and for the installed system.

Then select the appropriate **timezone:** Choose the country or region where you are located.

Specify whether your machine's **hardware clock** is set to local time or GMT.

Most PCs that also have other operating systems installed (like Microsoft Windows) use local time.

Machines that have

Basic Configuration

Keyboard layout

English (US)
Finnish
French
French (Switzerland)
German
German (Swiss)
Greek
Hungarian
Italian
Japanese
Norwegian
Polish
Portugese

Timezone

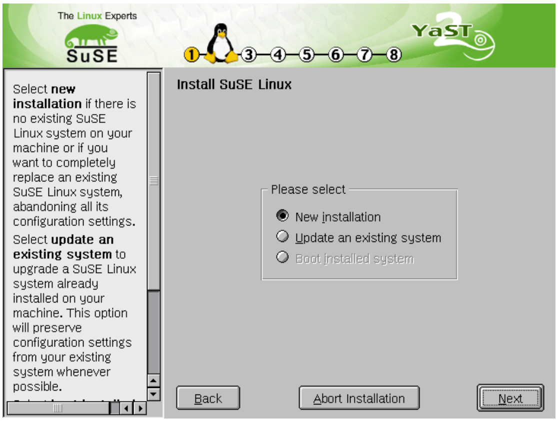
US/Alaska
US/Aleutian
US/Arizona
US/Central
US/East-Indiana
US/Eastern
US/Hawaii
US/Indiana-Starke
US/Michigan
US/Mountain
US/Pacific
US/Samoa

Keyboard test:

Hardware clock set to: local time

Back | Abort Installation | Next

Figure 2.5: Selecting the keyboard layout and time zone.



Select **new installation** if there is no existing SuSE Linux system on your machine or if you want to completely replace an existing SuSE Linux system, abandoning all its configuration settings.

Select **update an existing system** to upgrade a SuSE Linux system already installed on your machine. This option will preserve configuration settings from your existing system whenever possible.

Install SuSE Linux

Please select

● New installation
○ Update an existing system
○ Boot installed system

Back | Abort Installation | Next

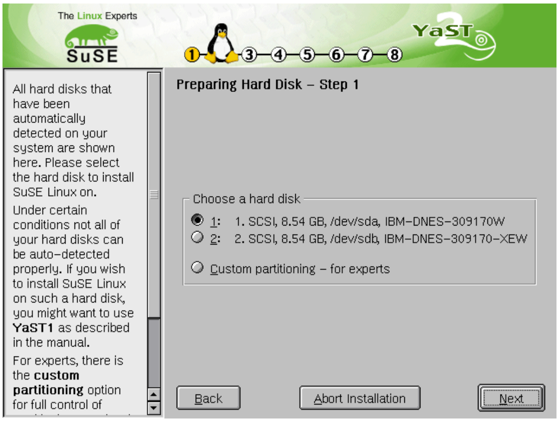
Figure 2.6: Selection installation type

Note

How the installation proceeds will depend on the installation method you have chosen and the components in your computer. For this reason, certain dialogs may not appear in the following pages on your screen.

2.1.9 Selecting a Hard Disk

In the new installation, only one dialog will be shown for the hard disk (Figure 2.7 on the next page). All the hard disks detected on your computer will be shown. Choose the disk on which you would like to install SuSE Linux. There is not much risk in selecting the hard disk. No existing data will be deleted and no changes will be made to your system at this point. This option is provided in the menu, '`Custom partitioning - for experts`'.



All hard disks that have been automatically detected on your system are shown here. Please select the hard disk to install SuSE Linux on.

Under certain conditions not all of your hard disks can be auto-detected properly. If you wish to install SuSE Linux on such a hard disk, you might want to use **YaST1** as described in the manual.

For experts, there is the **custom partitioning** option for full control of

Preparing Hard Disk – Step 1

Choose a hard disk

● 1: 1. SCSI, 8.54 GB, /dev/sda, IBM-DNES-309170W
○ 2: 2. SCSI, 8.54 GB, /dev/sdb, IBM-DNES-309170-XEW

○ Custom partitioning – for experts

Figure 2.7: Selecting the hard disk where SuSE Linux is to be installed

Preparation – Standard

YaST2 will now show all partitions present on the hard disk chosen (Figure 2.8 on the facing page). The free storage space is shown and automatically selected. If you want to assign more disk space to Linux, you can free up more disk space by following the order from the last to the first partition. If you have three partitions you can choose the first and the third for Linux and the second will be available for other operating systems to use.

If you click on '`Entire hard disk`', the entire hard disk and all the Linux partitions will be made available.

Caution: If you should decide to install Linux on the entire hard disk, all your previous data will be deleted!

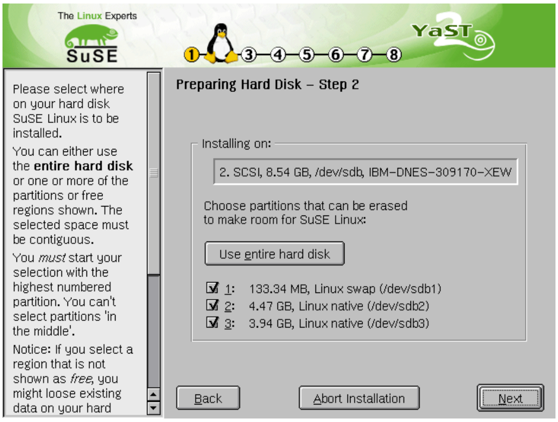
During the course of the installation process, YaST2 will verify

- whether there is sufficient storage for a minimum installation and
- whether the three standard partitions used by Linux can be created.

If this is not the case, you will prompted to alter your selection. If there is enough storage space, YaST2 will apply your settings so that the entire hard disk or selected partitions can be distributed over the three default partitions: boot, swap and root.

Caution

If you choose '`Use entire hard disk`', you will lose all the information on your hard disk!



Please select where on your hard disk SuSE Linux is to be installed.

You can either use the **entire hard disk** or one or more of the partitions or free regions shown. The selected space must be contiguous.

You *must* start your selection with the highest numbered partition. You can't select partitions 'in the middle'.

Notice: If you select a region that is not shown as *free*, you might loose existing data on your hard

Preparing Hard Disk – Step 2

Installing on:

2. SCSI, 8.54 GB, /dev/sdb, IBM-DNES-309170-XEW

Choose partitions that can be erased
to make room for SuSE Linux:

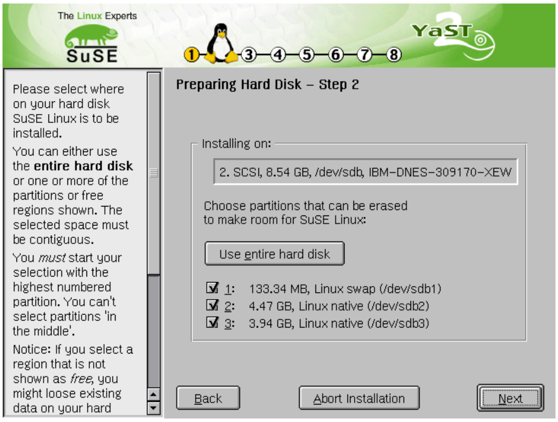
☑ 1: 133.34 MB, Linux swap (/dev/sdb1)
☑ 2: 4.47 GB, Linux native (/dev/sdb2)
☑ 3: 3.94 GB, Linux native (/dev/sdb3)

Figure 2.8: Selecting the partitions where SuSE Linux is to be installed

Preparation – Adjusting a Windows Partition

Windows 9x/ME is most likely located on the hard disk you selected. If this is the case, there is not enough space available for SuSE Linux. So, you are faced with two options: You can either completely remove Windows 9x/ME from your hard disk or you can shrink the Windows partition so that you will have enough space to install SuSE Linux (Figure 2.9).

If you decide to remove Windows 9x/ME, you should be aware that all the data will then be irrevocable lost.



Please select where on your hard disk SuSE Linux is to be installed.

You can either use the **entire hard disk** or one or more of the partitions or free regions shown. The selected space must be contiguous.

You *must* start your selection with the highest numbered partition. You can't select partitions 'in the middle'.

Notice: If you select a region that is not shown as *free*, you might loose existing data on your hard

Preparing Hard Disk – Step 2

Installing on:

2. SCSI, 8.54 GB, /dev/sdb, IBM-DNES-309170-XEW

Choose partitions that can be erased
to make room for SuSE Linux:

☑ 1: 133.34 MB, Linux swap (/dev/sdb1)
☑ 2: 4.47 GB, Linux native (/dev/sdb2)
☑ 3: 3.94 GB, Linux native (/dev/sdb3)

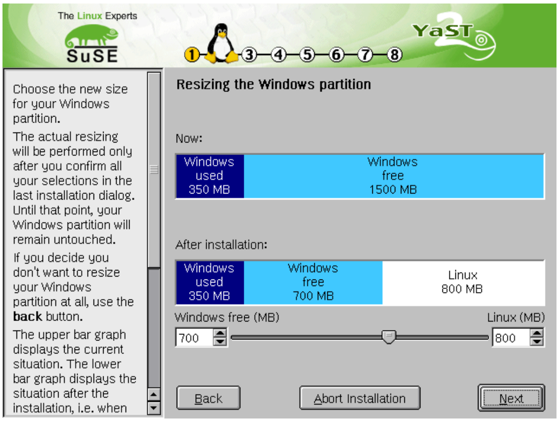
Figure 2.9: Available options for Windows partitions.

SuSE Linux can also coexist with other operating systems on your machine. If Microsoft Windows is already installed on your machine and the hard disk selected for SuSE Linux only has one partition with Windows (which is normally the case for pre-installed machines), you can shrink the part of your hard disk reserved for Windows without having to remove the operating system to install

SuSE Linux.

Before shrinking the Windows partition, you should boot Windows first. Run both the `scandisk` and `defrag` programs. Scandisk and Defrag will scan the files used in Windows and attach them to the beginning of the had disk before SuSE Linux subsequently scans the Windows part of the hard disk itself. This accelerates the reduction process.

YaST2 shows, in the first bar graph, how much memory Windows is currently occupying and how much hard disk memory is still available. The second bar graph suggests how the new hard disk should be repartitioned (Figure 2.10). You can accept this suggestion or, to the most extent, customize the size limits using the slide-control.



Choose the new size for your Windows partition.

The actual resizing will be performed only after you confirm all your selections in the last installation dialog. Until that point, your Windows partition will remain untouched.

If you decide you don't want to resize your Windows partition at all, use the **back** button.

The upper bar graph displays the current situation. The lower bar graph displays the situation after the installation, i.e. when

Resizing the Windows partition

Now:

Windows used 350 MB	Windows free 1500 MB

After installation:

Windows used 350 MB	Windows free 700 MB	Linux 800 MB

Windows free (MB): 700

Linux (MB): 800

Figure 2.10: Adjusting the Windows partition.

Note

If you have previously optimized a swap file in Windows, you should save it (fixed upper and lower limits of the swap file), as this range would otherwise be distributed over the entire hard disk space. Windows attributes swap files a special, invisible label. Defrag does not usually move such files because very old copy protection mechanisms in some older applications are based on this same label.

Caution

The Windows Resizer only works if your machine has a hard disk with exactly one Windows partition (this is the case for almost all pre-installed Windows machines). Otherwise, you will have to partition your hard disk by hand.

2.1.10 Scope of Installation

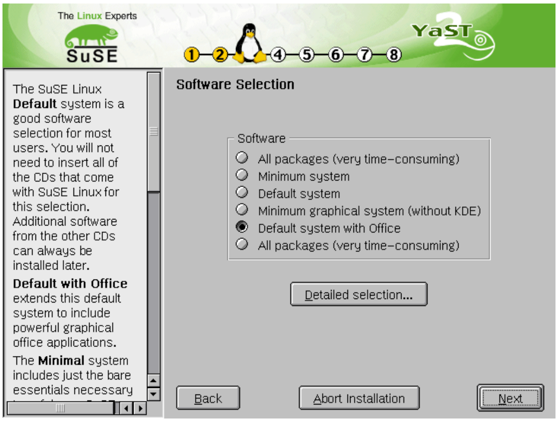
Specify which software is to be installed on your computer (Figure 2.11). SuSE Linux provides four base systems for this:

- **Minimum system**
  By selecting '`Minimum system`', a fully functional Linux operating system will be installed which is geared towards a text-based mode. This mode looks similar to DOS mode. This selection could be the right choice for you if you have limited space on your hard disk available, but would still like to use the basic functionality of SuSE Linux.

- **Default system**
  If you opt for '`Default system`', a well-balanced basic software environment will be installed on your system. This contains everything you might need at the beginning. It includes a graphical user interface, games, office packs, CD player, editors, etc. If you have no previous Linux experience, you should choose '`Default system`'.

- **Default system with Office**
  In the selection '`Default system with Office`', a wide array of office applications are available which incorporate word processing, spreadsheets, drawing tools and a daily planner which you can use to optimize your PC as an office workstation. If you still need a well-balanced software system for a fully functional office environment, you should choose this option.

- **All packages (very time-consuming)**
  If you would like the whole range of SuSE Linux software, choose '`All packages (very time-consuming)`'. This requires more than 6 GB free disk space. Note: This is called '`All packages (very time-consuming)`' because it does not make sense to install all the software from the 6 SuSE CD-ROMs, since some applications are mutually exclusive.



The SuSE Linux **Default** system is a good software selection for most users. You will not need to insert all of the CDs that come with SuSE Linux for this selection. Additional software from the other CDs can always be installed later.

Default with Office extends this default system to include powerful graphical office applications.

The **Minimal** system includes just the bare essentials necessary

Software Selection

Software
○ All packages (very time–consuming)
○ Minimum system
○ Default system
○ Minimum graphical system (without KDE)
● Default system with Office
○ All packages (very time–consuming)

Figure 2.11: Selecting the size of installation

Note

Along with the base systems, ‘`Minimum system`’ and ‘`Default system`’, you can choose additional components and software. To increase storage space for SuSE Linux later on, click on ‘`Back`’ until you get to item 2.1.9 page 13. There, you can redefine the new hard drive storage.

Selecting Software

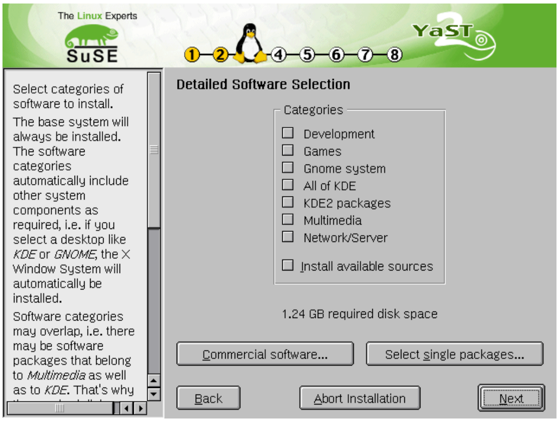
If you have made the ‘`Detailed selection`’, you can now choose different groups of software packages such as multi-media or games to be installed in addition to the previously selected software packages (Figure 2.12).

Choose the desired software from the entries provided.

At this stage in the installation, YaST2 offers two further options to choose from:

- Commercial software
  If you are interested in the Commercial Linux applications, go to Section 2.1.10.
- Selecting individual packages
  You may also select individual software packages within a group. Go to Section 2.1.10 on the next page.

In both cases, YaST2 will return you to the detailed selection.



Select categories of software to install.
The base system will always be installed. The software categories automatically include other system components as required, i.e. if you select a desktop like *KDE* or *GNOME*, the X Window System will automatically be installed.
Software categories may overlap, i.e. there may be software packages that belong to *Multimedia* as well as to *KDE*. That's why

Detailed Software Selection

Categories

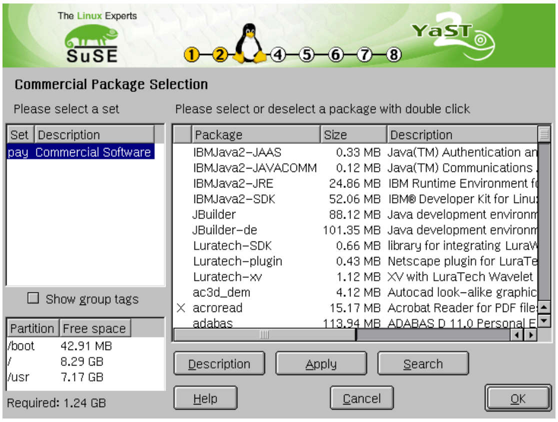
- □ Development
- □ Games
- □ Gnome system
- □ All of KDE
- □ KDE2 packages
- □ Multimedia
- □ Network/Server
- □ Install available sources

1.24 GB required disk space

Figure 2.12: Selecting additional components

Commercial Software

According to the SuSE philosophy, all commercial software products are listed in separate series. Here you can find a wide variety of applications such as office products, games and much more. It is well worth having a closer look! If you have selected the option ‘Commercial Software’ under ‘Detailed Software Selection’, you will see a screen as depicted in Figure 2.13 which allows you to select the commercial software packages to be installed. Click on the desired software package to select it. If you change your mind about installing one of the packages selected, simply click on it again to deselect it.



Commercial Package Selection

Please select a set

Set	Description
pay	Commercial Software

□ Show group tags

Partition	Free space
/boot	42.91 MB
/	8.29 GB
/usr	7.17 GB

Required: 1.24 GB

Please select or deselect a package with double click

	Package	Size	Description
	IBMJava2-JAAS	0.33 MB	Java(TM) Authentication an
	IBMJava2-JAVACOMM	0.12 MB	Java(TM) Communications .
	IBMJava2-JRE	24.86 MB	IBM Runtime Environment fo
	IBMJava2-SDK	52.06 MB	IBM® Developer Kit for Linu:
	JBuilder	88.12 MB	Java development environm
	JBuilder-de	101.35 MB	Java development environm
	Luratech-SDK	0.66 MB	library for integrating LuraW
	Luratech-plugin	0.43 MB	Netscape plugin for LuraTe
	Luratech-xv	1.12 MB	XV with LuraTech Wavelet
	ac3d_dem	4.12 MB	Autocad look-alike graphic
×	acroread	15.17 MB	Acrobat Reader for PDF file:
	adabas	113.94 MB	ADABAS D 11.0 Personal E

Figure 2.13: Selecting commercial software

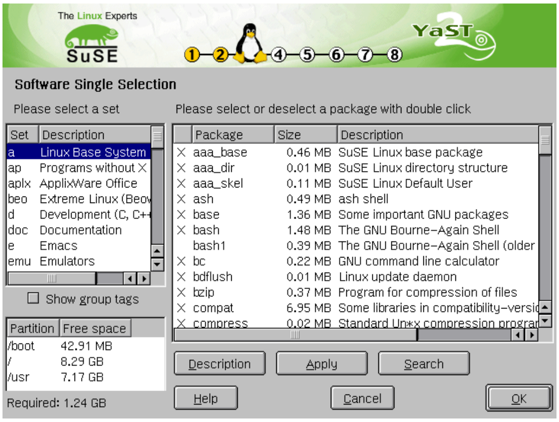
> ### Caution
>
> Please note that some of the software packages which can be installed here do not correspond to the full versions which can be purchased separately. Some offer limited functionality or can be used only for a certain length of time while others require registration. The relevant license agreements provide further information on the usage of these products.

Selecting Individual Applications

As with the commercial packages, the individual software selection allows you to select additional software packages. Under ‘Detailed Software Selection’, if you have chosen the option ‘Select single packages’, you will get to a screen as shown in Figure 2.14 on the next page. Click on the software packages you want to select. If you change your mind and decide to install one of the packages you selected, simply click on it again to deselect it.

> ### Caution
>
> In some cases, deselecting certain packages can lead to reduction of your system’s functionality.

Click on ‘Next’ or ‘OK’ to continue the installation.



Software Single Selection

Please select a set

Set	Description
a	Linux Base System
ap	Programs without X
aplx	ApplixWare Office
beo	Extreme Linux (Beow
d	Development (C, C++
doc	Documentation
e	Emacs
emu	Emulators

□ Show group tags

Partition	Free space
/boot	42.91 MB
/	8.29 GB
/usr	7.17 GB

Required: 1.24 GB

Please select or deselect a package with double click

	Package	Size	Description
X	aaa_base	0.46 MB	SuSE Linux base package
X	aaa_dir	0.01 MB	SuSE Linux directory structure
X	aaa_skel	0.11 MB	SuSE Linux Default User
X	ash	0.49 MB	ash shell
X	base	1.36 MB	Some important GNU packages
X	bash	1.48 MB	The GNU Bourne–Again Shell
	bash1	0.39 MB	The GNU Bourne–Again Shell (older
X	bc	0.22 MB	GNU command line calculator
X	bdflush	0.01 MB	Linux update daemon
X	bzip	0.37 MB	Program for compression of files
X	compat	6.95 MB	Some libraries in compatibility–versio
X	compress	0.02 MB	Standard Un*x compression program

Figure 2.14: Selecting individual applications

2.1.11 LILO – Introduction

In order to be able to start SuSE Linux or any other operating system on your computer, you will need a program to help determine whether you want to use SuSE Linux or another operating system when starting (or booting) the machine. One such program is the boot manager LILO, the “**LI**nux **LO**ader”.

YaST2 rrecognizes if you are using another operating system other than SuSE Linux. Depending on your machine’s environment, there are four alternative installation steps.

LILO – Multiple Operating Systems

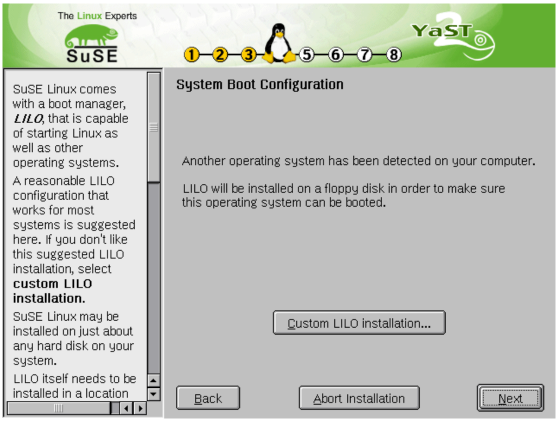
If you already have another operating system installed, YaST2 will suggest that you save the LInux LOader to a floppy. This way, you can start Linux by first inserting this disk into the drive before switching on your computer. If the disk is not in the drive, the other operating system will start. Later, you can alternatively select the LILO expert installation in YaST2 and configure LILO so that you can select which operating system should be started at system start.

LILO – SuSE Linux Only

If SuSE Linux is to be the only operating system existing on your computer, the dialog shown in Figure 2.16 on the facing page will appear.

LILO – Multiple Hard Disks

If you have two or more hard disks on your computer, YaST2 can not explicitly state on its own, from which hard disk the operating system should be booted. The hard disks will be enumerated by YaST2 and the size of the disk space as



SuSE Linux comes with a boot manager, *LILO*, that is capable of starting Linux as well as other operating systems.

A reasonable LILO configuration that works for most systems is suggested here. If you don't like this suggested LILO installation, select **custom LILO installation.**

SuSE Linux may be installed on just about any hard disk on your system.

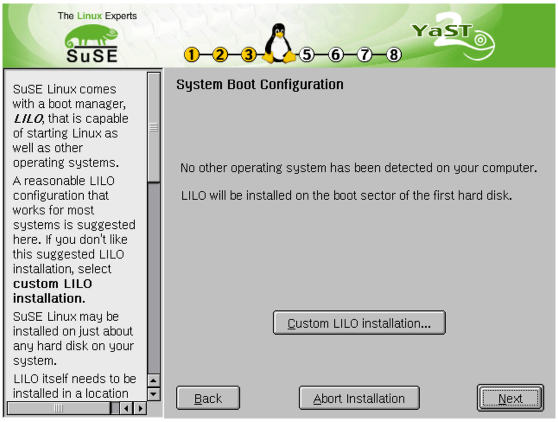
LILO itself needs to be installed in a location

System Boot Configuration

Another operating system has been detected on your computer.

LILO will be installed on a floppy disk in order to make sure this operating system can be booted.

Figure 2.15: LILO installation on disk



SuSE Linux comes with a boot manager, *LILO*, that is capable of starting Linux as well as other operating systems.

A reasonable LILO configuration that works for most systems is suggested here. If you don't like this suggested LILO installation, select **custom LILO installation.**

SuSE Linux may be installed on just about any hard disk on your system.

LILO itself needs to be installed in a location

System Boot Configuration

No other operating system has been detected on your computer.

LILO will be installed on the boot sector of the first hard disk.

Figure 2.16: LILO – SuSE as single operating system

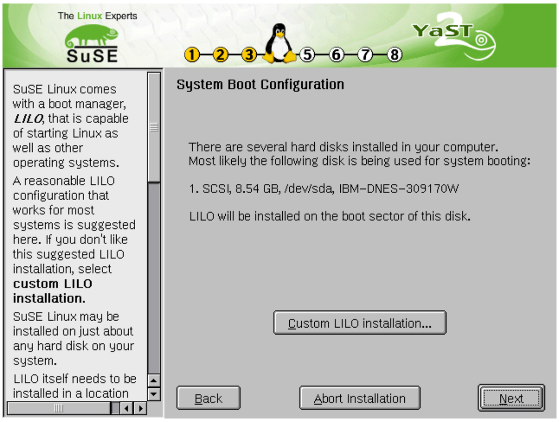
well as the drive label will be shown in Linux (e. g. /dev/hda) (Figure 2.17 on the next page). Drive labels such as C:, D:, E: ... are irrelevant in Linux.

Note

YaST2 suggests a hard disk to select. If you encounter trouble when restarting the system, you should call up YaST2 again and choose another hard disk in the item '`Other configuration`'.

LILO – When All Else Fails

In rare cases, the dialog will appear as shown in Figure 2.18 on the following page. In such cases, LILO will not automatically be installed to your hard disk.



SuSE Linux comes with a boot manager, *LILO*, that is capable of starting Linux as well as other operating systems.

A reasonable LILO configuration that works for most systems is suggested here. If you don't like this suggested LILO installation, select **custom LILO installation.**

SuSE Linux may be installed on just about any hard disk on your system.

LILO itself needs to be installed in a location

System Boot Configuration

There are several hard disks installed in your computer.
Most likely the following disk is being used for system booting:

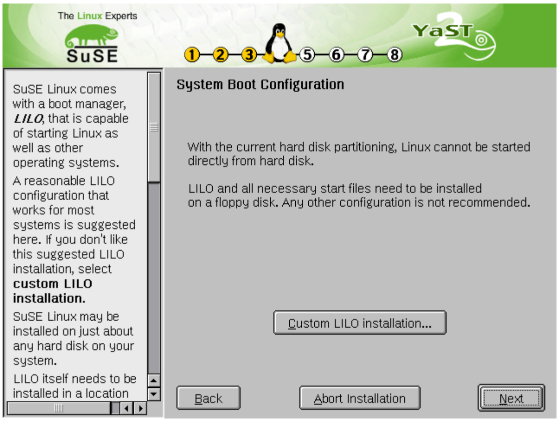
1. SCSI, 8.54 GB, /dev/sda, IBM-DNES-309170W

LILO will be installed on the boot sector of this disk.

Figure 2.17: LILO – Installation in the MBR

SuSE Linux can, however, be booted from a start disk.

YaST2 will not change the related start settings, but will rather suggest that you save the LInux LOader to a floppy. This way, you can start Linux by first inserting this floppy into the disk drive before switching on your computer. If the floppy is not in the drive, your previous operating system will start. Once you are more comfortable using Linux, you can call up '`Other configuration`' in YaST2 and change these settings. Then, you will have the option of selecting an operating system at system start via LILO.



SuSE Linux comes with a boot manager, *LILO*, that is capable of starting Linux as well as other operating systems.

A reasonable LILO configuration that works for most systems is suggested here. If you don't like this suggested LILO installation, select **custom LILO installation.**

SuSE Linux may be installed on just about any hard disk on your system.

LILO itself needs to be installed in a location

System Boot Configuration

With the current hard disk partitioning, Linux cannot be started directly from hard disk.

LILO and all necessary start files need to be installed on a floppy disk. Any other configuration is not recommended.

Figure 2.18: LILO – The most complicated scenario

LILO – Other Configuration

If you chose '`Other configuration`', a screen will appear as shown in Figure 2.19 on the next page.

In this dialog, you can decide for yourself where LILO is to be installed.

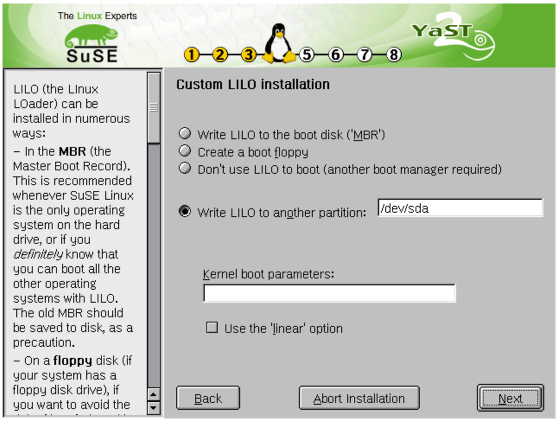
Four possibilities are open to you:

1. Install to /dev/hda (C:) (in the MBR of your hard disk)
2. Create a boot floppy
3. Install a boot manager other than LILO
4. Install to another partition

Install SuSE Linux as a single operating system on your machine, then select Option 1: to C: . With this, LILO will be installed to the Master Boot Record (MBR) of your hard disk. Also choose this option if you want to use LILO as the boot manager for several operating systems. Make sure, however, that your operating system can be booted by LILO (usually MS-DOS and Windows 95/98).

If you are using several operating systems which you are not sure can be booted by LILO, or if you do not want to make any changes to the previous start mechanism unchanged, then choose the option '`Create boot floppy`'. This way, if need be, you can boot SuSE Linux from disk.

If you have already installed a boot manager and want to include SuSE Linux in this boot manager, select '`Do not install LILO`'. After installing SuSE Linux you will have to reconfigure the existing boot manager and integrate SuSE Linux into the boot procedure.



LILO (the LInux LOader) can be installed in numerous ways:

– In the **MBR** (the Master Boot Record). This is recommended whenever SuSE Linux is the only operating system on the hard drive, or if you *definitely* know that you can boot all the other operating systems with LILO. The old MBR should be saved to disk, as a precaution.

– On a **floppy** disk (if your system has a floppy disk drive), if you want to avoid the

Custom LILO installation

○ Write LILO to the boot disk ('MBR')
○ Create a boot floppy
○ Don't use LILO to boot (another boot manager required)

● Write LILO to another partition: /dev/sda

Kernel boot parameters:

□ Use the 'linear' option

Figure 2.19: LILO – Expert mode

Tip

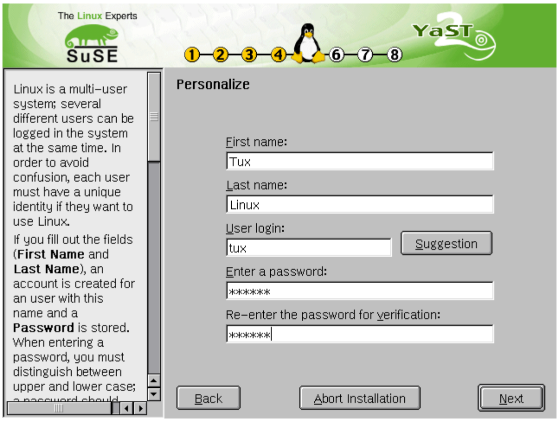
If you install LILO on a boot floppy, do not change anything with your previous boot mechanism and you can still start SuSE Linux from the floppy at any time. The option '`Create boot floppy`' is thus the safest alternative for using additional operating systems.

2.1.12 Username and Password

Linux is an operating system which allows several users to work on the same system at the same time. For it to function smoothly, each user needs a user account which allows them to log on to the system and then to log out again when they are finished. Setting up user accounts ensures excellent operating security. Normal users will not be able to change or delete files needed for the system to work properly. Similarly, a user's personal data may not be accessed, modified, or deleted by other users. Every user can set up his or her own working environment with their favorite applications and settings. Each time they log on to the Linux system, they will find the environment unchanged.

You can create such a user account yourself using the dialog shown in Figure 2.20. You will be asked to enter your first name and your last name. You will also need to specify your username (login name). If you cannot think of a suitable username, you can click on the '`Suggestion`' button and the system will automatically generate a username for you.

Finally, you will need to enter a password for your username which you must repeat to confirm. The username tells the system who you are and the password verifies your identity.



Linux is a multi-user system; several different users can be logged in the system at the same time. In order to avoid confusion, each user must have a unique identity if they want to use Linux.

If you fill out the fields (**First Name** and **Last Name**), an account is created for an user with this name and a **Password** is stored. When entering a password, you must distinguish between upper and lower case; a password should

Personalize

First name: Tux

Last name: Linux

User login: tux

Enter a password: ******

Re-enter the password for verification: ******

Figure 2.20: Entering username and password

Caution

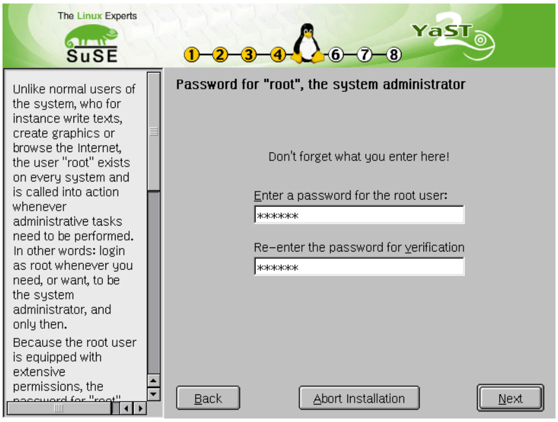
You should be sure to memorize your username and password, as you will need this information every time you log in.

A password should be between 5 - 8 characters long. It could be as long as 128 characters, but only the first 8 characters are shown. Linux distinguishes between lower and uppercase letters in the password. Accented characters are not allowed. Special characters may be used, however (e. g. *, ., # , ;) as well as the digits 0-9.

2.1.13 Root Password

root is the name of the superuser, or the system administrator. root is permitted to do all the things that the normal user is not permitted to do. The superuser may make changes to the system such as installing new applications or setting up new hardware. If someone has forgotten their password or has problems with software, root is able to help them. As a general rule, you should only log in as root if you wish to carry out administrative tasks such as system maintenance or repairs. But root is quite risky for everyday use, as root can delete files which can not be subsequently recovered, or which could disable the system.

This is why you already set your login in the previous screen, so that you can work on your computer without risking damage to the system. When setting the password for root, the same rules apply as with every other login. For verification purposes, the password has to be entered twice (Figure 2.21). Be particularly careful not to forget the root password. It can no longer be retrieved at any later point in time.



Unlike normal users of the system, who for instance write texts, create graphics or browse the Internet, the user "root" exists on every system and is called into action whenever administrative tasks need to be performed. In other words: login as root whenever you need, or want, to be the system administrator, and only then.

Because the root user is equipped with extensive permissions, the password for "root"

Password for "root", the system administrator

Don't forget what you enter here!

Enter a password for the root user: ******

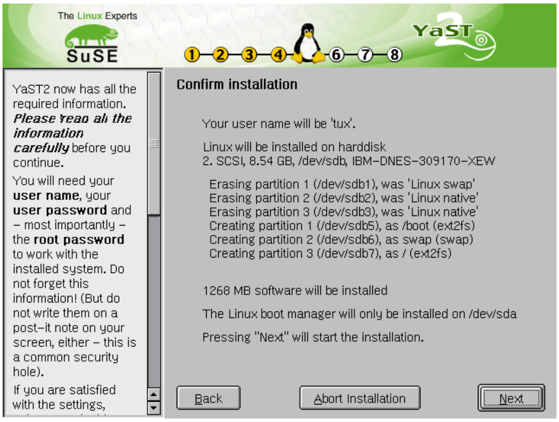
Re-enter the password for verification ******

Figure 2.21: Setting the password for user root

> **Caution**
>
> The user root has all the permissions needed to make changes to the system. If you want to carry out such tasks, the root password is required. You cannot carry out any administrative tasks without this password!

2.1.14 Let's Go!

In the following dialog box (Figure 2.22 on the following page), you will see your previously chosen settings listed. You can also '`Abort installation`' here. The installation of SuSE Linux will then be ended and your system will remain unchanged. If you want to change some of your settings, you can click repeatedly on ‘Back’ until you’ve reached the dialog box where you want to make your changes. If you click on ‘Next’, however, a dialog box will appear, asking you if you are sure you want to proceed with the installation. If you answer ‘Yes - install’, the installation will begin. If you want to save your selections for later retrieval, click on ‘Save settings to floppy’ and all the installation settings will be saved to a disk.



YaST2 now has all the required information. *Please read all the information carefully* before you continue.

You will need your **user name**, your **user password** and – most importantly – the **root password** to work with the installed system. Do not forget this information! (But do not write them on a post–it note on your screen, either – this is a common security hole).

If you are satisfied with the settings,

Confirm installation

Your user name will be 'tux'.

Linux will be installed on harddisk
2. SCSI, 8.54 GB, /dev/sdb, IBM–DNES–309170–XEW

Erasing partition 1 (/dev/sdb1), was 'Linux swap'
Erasing partition 2 (/dev/sdb2), was 'Linux native'
Erasing partition 3 (/dev/sdb3), was 'Linux native'
Creating partition 1 (/dev/sdb5), as /boot (ext2fs)
Creating partition 2 (/dev/sdb6), as swap (swap)
Creating partition 3 (/dev/sdb7), as / (ext2fs)

1268 MB software will be installed

The Linux boot manager will only be installed on /dev/sda

Pressing "Next" will start the installation.

Figure 2.22: List of changes made

Caution

All data on the partitions you have specified for SuSE Linux will, in the next step, be deleted irrevocably. If you have chosen the entire hard disk, all other operating systems and data here will be erased.

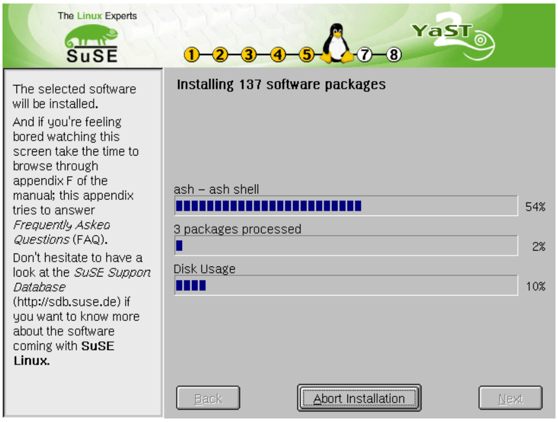
2.1.15 Preparing the Hard Disk

YaST2 will now begin its work. YaST2 will create the selected partitions and format them. Depending on your system configuration, this may take some time.

2.1.16 Installation of Packages

Once you have started the installation process, the selected packages of the Linux base system are copied from CD or DVD and written to your hard disk. On this screen, you can monitor the progress of the various tasks (Figure 2.23 on the next page).

Depending on the system configuration, scope of installation and size of the hard disk, the installation can be somewhat time-consuming.



The selected software will be installed.

And if you're feeling bored watching this screen take the time to browse through appendix F of the manual; this appendix tries to answer *Frequently Asked Questions* (FAQ).

Don't hesitate to have a look at the *SuSE Support Database* (http://sdb.suse.de) if you want to know more about the software coming with **SuSE Linux.**

Installing 137 software packages

ash – ash shell 54%

3 packages processed 2%

Disk Usage 10%

Figure 2.23: Package installation

2.1.17 Starting the Base System

Once all the packages on the first CD of SuSE Linux have been copied to the hard disk, a basic Linux system will be started. You will be notified by a corresponding message. Please confirm this message so that the installation can be continued.

2.1.18 And next...

After starting the base system, the additional packages you chose will be installed as well. Depending on the amount of software you chose, you may not necessarily need all the CDs. After completing these tasks, all the software packages will be installed on your machine.

Note

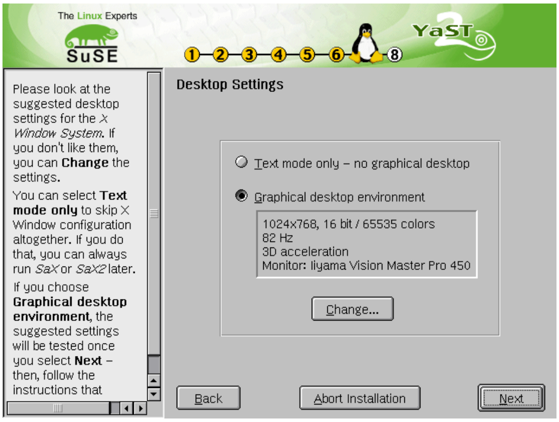
Depending on the configuration you chose for the installation of LILO, you may be prompted to insert a disk to create a boot floppy. Please note that in doing so, all the data located on the medium will be deleted.

2.1.19 Monitor Settings

In the following dialog screen (Figure 2.24 on the following page) you can define whether, in the future, SuSE Linux should be operated in '`Text mode only`' or in a '`Graphical desktop environment`'.

When you click on '`Change`', you will have the option of configuring the graphical interface (Figure 2.25 on the next page).

You can set the screen resolution and color depth for the graphical mode. You can even define the image repetition rate.



Please look at the suggested desktop settings for the *X Window System*. If you don't like them, you can **Change** the settings.

You can select **Text mode only** to skip X Window configuration altogether. If you do that, you can always run *SaX* or *SaX2* later.

If you choose **Graphical desktop environment**, the suggested settings will be tested once you select **Next** – then, follow the instructions that

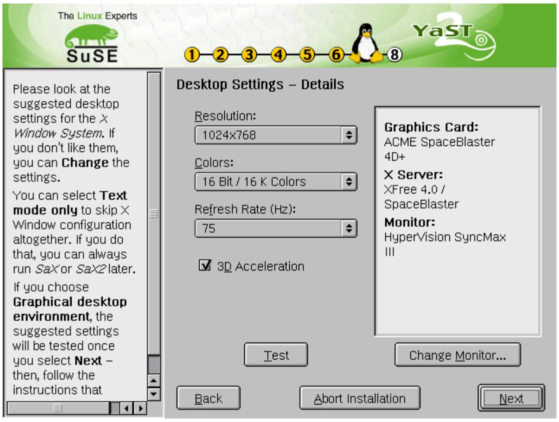
Desktop Settings

○ Text mode only – no graphical desktop

● Graphical desktop environment

1024x768, 16 bit / 65535 colors
82 Hz
3D acceleration
Monitor: Iiyama Vision Master Pro 450

Figure 2.24: Monitor settings



Please look at the suggested desktop settings for the *X Window System*. If you don't like them, you can **Change** the settings.

You can select **Text mode only** to skip X Window configuration altogether. If you do that, you can always run *SaX* or *SaX2* later.

If you choose **Graphical desktop environment**, the suggested settings will be tested once you select **Next** – then, follow the instructions that

Desktop Settings – Details

Resolution: 1024x768

Colors: 16 Bit / 16 K Colors

Refresh Rate (Hz): 75

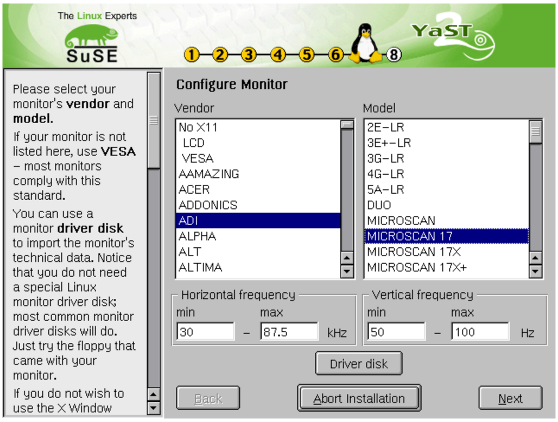
☑ 3D Acceleration

Graphics Card:
ACME SpaceBlaster 4D+
X Server:
XFree 4.0 / SpaceBlaster
Monitor:
HyperVision SyncMax III

Figure 2.25: Changing the settings for the graphical interface

If you have a graphics card with a 3D chip, you can select the option '`3D acceleration`'. By clicking on the '`Test`' button, you can test the resolution you have selected. The installation program will issue a message telling you that the screen will now switch over to the new resolution. If you don't see a steady screen, please stop the test immediately by pressing ESC. Clicking on '`Change Monitor`' will take you to the screen shown in Figure 2.26 on the facing page.

Some technical data regarding your selected model, the horizontal (HSync) and the vertical (VSync) frequency deflection rates will appear in the bottom portion of the screen. If the preferred model is not included in the list, you can manually enter the data in the entry fields, or choose pre-defined settings (VESA modes). Please use the relevant values listed in your monitor manual. Otherwise, you can use a driver floppy. To do this, click on '`Driver disk`'. Insert the disk into the drive and confirm with '`OK`'. If no file could be found or if the floppy is not readable, you will receive the respective warning. Then, the monitor data will



Please select your monitor's **vendor** and **model.**

If your monitor is not listed here, use **VESA** – most monitors comply with this standard.

You can use a monitor **driver disk** to import the monitor's technical data. Notice that you do not need a special Linux monitor driver disk; most common monitor driver disks will do. Just try the floppy that came with your monitor.

If you do not wish to use the X Window

Configure Monitor

Vendor: No X11, LCD, VESA, AAMAZING, ACER, ADDONICS, ADI, ALPHA, ALT, ALTIMA

Model: 2E–LR, 3E+–LR, 3G–LR, 4G–LR, 5A–LR, DUO, MICROSCAN, MICROSCAN 17, MICROSCAN 17X, MICROSCAN 17X+

Horizontal frequency: min 30 – max 87.5 kHz

Vertical frequency: min 50 – max 100 Hz

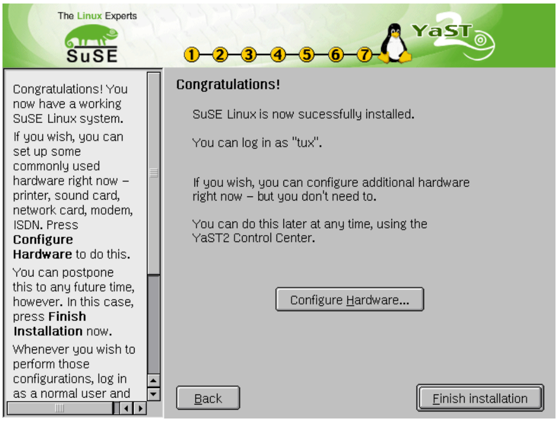
Figure 2.26: Selecting the monitor model

appear in the selection list.

To be on the safe side, you can also choose a standard resolution at this point. Select the item VESA and select `640 x 480` from the list.

2.1.20 Configuration

Once your graphics card has been configured, you will see the screen as shown in Figure 2.27. You can start the hardware configuration by clicking on '`Configure Hardware`'.



Congratulations! You now have a working SuSE Linux system.

If you wish, you can set up some commonly used hardware right now – printer, sound card, network card, modem, ISDN. Press **Configure Hardware** to do this.

You can postpone this to any future time, however. In this case, press **Finish Installation** now.

Whenever you wish to perform those configurations, log in as a normal user and

Congratulations!

SuSE Linux is now sucessfully installed.

You can log in as "tux".

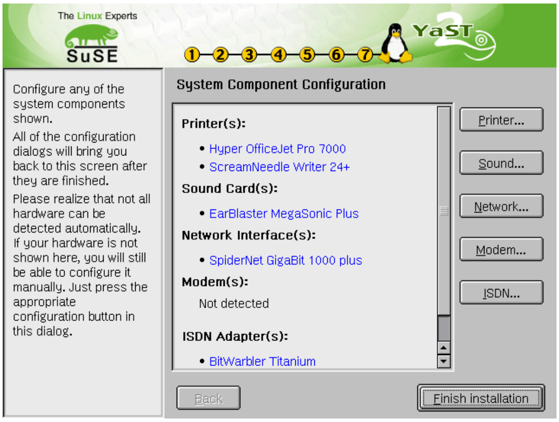
If you wish, you can configure additional hardware right now – but you don't need to.

You can do this later at any time, using the YaST2 Control Center.

Figure 2.27: Completing installation

YaST2 will show you the hardware on your computer which can be configured (Figure 2.28 on the following page). You can either carry out the configuration now or later. We suggest that you configure this later, once you have become familiar with SuSE Linux.

Now, please click on ‘`Finish installation`’.



Configure any of the system components shown.

All of the configuration dialogs will bring you back to this screen after they are finished.

Please realize that not all hardware can be detected automatically. If your hardware is not shown here, you will still be able to configure it manually. Just press the appropriate configuration button in this dialog.

System Component Configuration

Printer(s):

- Hyper OfficeJet Pro 7000
- ScreamNeedle Writer 24+

Sound Card(s):

- EarBlaster MegaSonic Plus

Network Interface(s):

- SpiderNet GigaBit 1000 plus

Modem(s):

Not detected

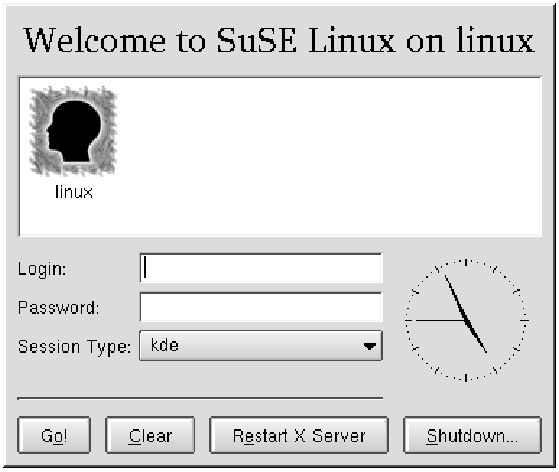
ISDN Adapter(s):

- BitWarbler Titanium

Figure 2.28: Configuring the system components

2.1.21 Graphical Login

SuSE Linux is now installed and configured so that you can log on to your system. Your monitor will now display the graphical login which you can see in Figure 2.29. Enter the username you specified earlier as well as your login password to log on to your system.



Welcome to SuSE Linux on linux

linux

Login:

Password:

Session Type: kde

Figure 2.29: Logging on to the system

2.2 Starting SuSE Linux

After installation, the question remains as to how you want to boot your system normally (Booting).

The following section describes the options you have for booting your SuSE Linux system. The best boot method for you will depends on the intended use of your Linux system.

Boot disk: You can boot Linux from a *Boot disk* ("Boot floppy"). This choice will always work (as long as the boot disk does not get corrupted) and it is easy. The boot disk may have been created during the installation.

loadlin: The loadlin boot option can be configured as follows:

- The computer must be running either in DOS Real mode or have a VCPI server in virtual 8086 mode [1] In other words: this method will *not* function in Unix, OS/2, Windows NT or in a DOS window on a Windows 95/98 machine. It does, however, function well from MSDOS or from Windows 95/98 in DOS mode.
- Your computer must have enough DOS memory available: There must be 128 KB available below the 640 KB limit, the rest can be on extended/EMS/XMS memory.

Loadlin is fairly labor intensive to set up, but it can be easily integrated into the Windows 95/98 boot menus. This requires a manual editing of configuration files. One of the biggest advantages of loadlin is that nothing gets installed into the MBR (*Master Boot Record*) of the hard disk. To other operating systems, Linux partitions will appear as an unknown type.

To install loadlin, you need to know a little about Linux and DOS. You should also be able to create configuration files with an editor. You can find details of this in Section 4.9 page 117. If you make a mistake in the Windows95/98 boot menus, this could cause problems. In the event of an extreme error, you may loose access to your Windows hard drive. *Before* you start modifying your Windows boot menu, make sure that you can boot your Windows operating system with a boot floppy.

LILO: The universal and technically elegant solution for booting your system is LILO. The LILO boot menu can be configured to give you a choice of operating systems to boot, before anything is booted. With YaST it is relatively easy to configure and install LILO (see Section 3.6.2 page 81). LILO must be installed in the boot sector of your hard disk, and this is not without some risk. To install it correctly you will need to know a bit more about the boot process than the average user. You should be comfortable enough with the main LILO configuration file to edit it. You are well advised to learn how to uninstall LILO in the event that you have difficulties. Details on LILO and the boot process can be found in Section 4.3 page 95. LILO is considered the best boot method, but we should warn you here that it is a bit more difficult to configure than a simple boot disk.

[1] A VCPI server is accessed by the program emm386.exe.

Caution

There are BIOS variants which check the structure of the boot sector (MBR), and after a LILO installation erroneously display a virus warning. This problem can be easily removed by entering the BIOS and looking for corresponding adjustable settings; for example, you should switch off ‘`virus protection`’. You can switch this option back on again later; this feature is unnecessary, however, if Linux is the only operating system you are using.

A detailed discussion of various boot methods, especially of LILO and loadlin can be found in Chapter 4 page 93.

Other Boot Managers

Because of the increasing importance of Linux, some creators of commercial boot managers have included options for booting Linux in their products. Best known among boot managers are System Commander Deluxe and Partition Magic. In addition to help screens at boot time, many of these packages offer a lot of functionality. For example it is possible to extend existing FAT32 partitions or to change FAT16 partitions into FAT32. You can *not* find these programs on the installation CD’s and we do *not* offer *Installation support* for these products!

2.3 Special Installations

2.3.1 Installation Without a Supported CD-ROM Drive

What do you do if a standard installation via the CD-ROM drive is not possible? Your CD-ROM drive might not be supported if it is an older, "proprietary" model. Or it might be your second computer, (a Notebook, for example) which might not even have a CD-ROM drive, but instead has an Ethernet adapter or a PLIP cable. . .

SuSE Linux provides ways of installing over a network connection to such a machine without using a supported CD-ROM drive:

NFS or FTP via Ethernet or via PLIP (Section 2.3.1)

are available to you.

Installation from a Source in the "Net"

We do not offer support for this method of installation. It is only recommended for experienced computer users.

What's it all about?

There is *no* CD-ROM installed on the machine on which you want to install Linux.

Still you are able to connect to a remote machine over the network that has an installed CD-ROM drive (and CD) or which has a copy of the CD on its hard disk.

In addition it is necessary to copy the files `.S.u.S.E-disk*` from the CD-ROM to the hard disk; this can be abbreviated in the following way:

```
earth: # cp /media/cdrom/.S* /emil
earth: # cp -a /media/cdrom/suse /emil
```

This "other" computer must "export" the directory in a suitable manner.

> **Tip**
>
> Alternatively it is sufficient to start the rescue system on the remote machine and directly export the CD 1.

Step by step

1. Start the installation of the client as specified in Section **??** page ??.
2. Continue with the installation as described in Section **??** page ??, but:
   - When you come to '`Kernel modules`', select '`Networking cards`' and load the necessary driver. This is not necessary if you are installing via PLIP.

  - When linuxrc asks you for a '`Source medium`', you should enter '`Network (NFS)`' and go through the menu for network configuration. An alternative is to install via FTP.

3. Finish up the installation as given in Section **??** page ??.

Possible problems

- The installation aborts before it has actually started, because the installation directory of the "other" machine wasn't exported with `exec` permissions. Correct this and start again.
- The server does not know the computer on which SuSE Linux is to be installed. Enter the name and IP address of the computer which is to be newly installed into the file `/etc/hosts` of the server.

2.3.2 Automatic Installation and Configuration of SuSE Linux

Why Automatization?

The automatic installation and configuration of Linux systems allows you to assemble a unified server landscape. Even for client installations, automatic installation is preferable – above a certain number – to a manual installation. This standardization involves the state of the system versions and the software, file system structures and configuration files. The automatization guarantees that an installation method which has once proven successful can be used on a computer at any time, without expert knowledge. In this way it is also possible to expand the server landscape without problems. This standardization is an obvious help for system administrators: if the same configuration structures are required, you don't need to search for configuration files on every server each time an installation is performed. The software behaves as expected, since the same versions are being used. Bugfixes will take effect across the whole server landscape.

All these effects improve the quality of production.

An overview of ALICE

ALICE ***A**utomatic **L**inux **I**nstallation and **C**onfiguration **E**nvironment* integrates the installation and essential areas of configuration.

Due to the CVS-based behavior of the configuration data, you can maintain this configuration data from various locations, and using different users.

It is possible to classify a computer into one or several different classes. This means that different computers can be kept as "similar" as possible; only really necessary data needs to be given in the computer-specific configuration file. To organize its structure, the configuration data is divided into three different directories. The `templates` directory contains a number of defaults, the `classes` directory contains class-specific configurations and the `info` directory contains host-specific information. The defaults can be overwritten by class values, and these, in turn, by special values assigned to the host.

ALICE is a growing project, i. e. if you do not find any modules for configuring software you are using, this can still be accomplished independently or in the framework of a project.

Further Information

Information on ALICE modules, setting up an installation server, the architecture of ALICE and a configuration session can be found in the documentation included in the package `alice`.

Automatic installation with YaST1

The "Automatic Installation" of YaST1 can also be used independently of the complete framework of the ALICE project; see also the hints in the article `file:/usr/share/doc/sdb/de/html/cg_autoinstall.html`.

2.4 Tips and Tricks

2.4.1 Creating Space for Linux (Partitioning)

What's it all about?

You need to prepare your hard disk for Linux partitions. Hopefully you have some time to read through this section carefully. It is recommended that you have some disks or tapes for a backup and a boot disk to boot your existing operating system.

Details

By partitioning, your hard disk can be separated into several independent parts. One reason for partitioning your hard disk might be the coexistence of several different operating systems with different file systems on your hard drive. hard disks are divided into cylinders. Each cylinder always contains exactly the same amount of tracks, cylinder 0 being the innermost.

How do you get space for your Linux partition?

- Individual partitions on the hard disk can be deleted but all data which resides on these partitions will be lost. This space is now available for new partitions (e. g., Linux). You could, however, split this partition into two or more partitions.

- Any partition can be assigned to a different file system. As with deleting, all data on these partitions will be lost.

- In MS-DOS or Windows 95/98 you can reduce the last partition of the hard disk in size without losing files in it; you must ensure, with a defragmentation program, that all files are located at the beginning of the partition. If you just have *one* MS-DOS or Windows partition you can make space in this way for Linux partitions very simply. After defragmenting you can, with special programs – for example, the Freeware program, fips, reduce the cylinder upper limit, making the partition smaller. You can find fips on CD 1 in the directory `dosutils`.
- If you are using MS-DOS or Windows, you can minimize the last partition without losing data.

  If all data resides on low-numbered cylinders, special programs (e. g., fips) can decrease the upper cylinder limit to reduce the size of the last partition leading to the possibility of creating a new partition for Linux. Using a defragmentation program ensures, that all data is moved to the beginning of the partition.
- The easiest way is to buy a new hard disk. This, of course, will cost you some money.

> Caution
>
> Every change of the partition tables must be done with extreme caution! Please read the documentation of the corresponding programs! Severe problems can occur while changing partition tables; you can even lose all your data. We at SuSE cannot be held responsible for this! It is strongly recommended that you have at least a boot disk and a backup of your important data.

Step by step

This is how to proceed to partition your hard disk:

1. If you don't know already, you have to find out how many partitions are on your hard disk and the size of the partitions. Use fdisk of your OS for this task.
2. Determine how many partitions you really need and which size you are going to assign them. Information on this may be found in Section 2.5 page 43 and in Section 2.6 page 45.
3. Write down the partition data; you're going to need it later during the installation process.
4. A backup of your hard disk at this point is highly recommended! If you don't have a backup tape or streamer installed and don't want to back up everything onto floppies (which is a tedious job), at least make a backup of the most important data and files (e. g., `autoexec.bat`, `config.sys`,

`*.ini`, etc.). Create a boot disk for your OS **and** make sure you can boot with it! There are several programs that might come in handy on your boot disk, such as fdisk, an editor, a formatting program and a backup program.

5. The next step depends on your system configuration.

DOS/Windows 95/98, one partition on your hard disk, and no Backup

You have to reduce the size of your partition without deleting data. Move all data to the beginning of your partition using a tool such as defrag (MS-DOS 6 or Windows 95/98).

The defragmentation program doesn't normally move hidden or system files because sometimes they are write-protected and have a defined place on the hard disk; moving them could have adverse effects. If you are sure that no such data is on your hard disk you can deactivate the attributes `hidden` or `system` for the necessary files or, depending on which defragmentation program you are using, you can "force" the defragmentation of such files.

Please keep in mind that even the Windows swap file is a hidden file. If it's in your way, you have to deselect it in Windows in "disable virtual memory".

Once you have retrieved enough space, change to the directory `\dosutils\fips15` on your SuSE CD. There you will find fips.exe, with which you can make partitions smaller. Study the documentation included with fips carefully since changing partition tables can be quite tricky! The program fips.exe only runs on DOS, not on Windows; you have to exit Windows or (if using Windows 95/98) change into DOS mode. After running fips.exe, you will have a second partition which later will be divided into your Linux partitions.

Tip

fips.exe in `\dosutils\fips\fips20` is capable of shrinking `fat32` partitions. Please make a backup *before* trying out this fips version!

DOS/Windows 95/98 with several partitions or you have a Complete File Backup

You can delete your DOS partitions and afterwards install them again, reduced in size. You are going to lose all data on those partitions, so you have to have a complete backup available (file backup, no image backup!). Using fdisk, you delete all partitions and install the new ones. Next, you format these partitions, install your OS and copy all data from your backup back to disk (remember that you must have your backup program on your boot disk).

OS/2

Here, you have the following possibilities:

- Reduce your OS/2 partitions. To accomplish this, you must backup all data on your OS/2 partition, delete these partitions using OS/2's fdisk and reinstall them reduced in size. The space now available will be used later for your Linux partitions.
- If you plan to use Linux in future instead of OS/2, all you have to is change the partition type later on to Linux, or delete all OS/2 partitions and recreate them later as Linux partitions.
- Buy a new hard disk. Formatting and partitioning it will be done later with YaST.

6. Reboot your computer.
7. Check whether your old system is still running! Linux partitions on the freshly created new space will be installed later using YaST.

Possible problems

The following problems can occur:

- The partition might not be adequately reduced in size because defrag cannot move some files.
- In DOS/Windows your CD-ROM now has another hard disk identifier; in Windows 95/98 the computer hangs.

2.4.2 Booting with CD 2

As well as CD 1, the second CD is also bootable. While on CD 1 a 2.88 MB boot image is used, the second CD contains a traditional image of 1.44 MB in size.

You should use CD 2 when you know for certain that you can boot from CD, but when things don't work with CD 1 (the "fallback" solution). Unfortunately not every BIOS correctly recognizes the large images.

2.4.3 Creating a Boot Disk In DOS

Requirements

You need a formatted 3.5" floppy disk and a bootable 3.5" floppy drive. If you are working in Windows 95/98, you must launch setup from MS-DOS mode, and *not* from within a DOS window.

Additional information

CD 1 in the directory `/disks` contains a number of disk images. Such an image can be copied to a disk with the help of suitable auxiliary programs, this disk is then called a boot disk. Also on these disk images are the "loader", Syslinux and program linuxrc. Syslinux allows you to select a specific kernel for the booting process, and to add parameters for your hardware, if necessary. The program linuxrc supports the loading of kernel modules specially for your hardware and then starts the installation.

Normally the SuSE boot disk supplied can be used to boot with. Only for exotic hardware not supported by the modularized kernel of this boot disk or if you download a disk image from the Internet (for example from `ftp://ftp.suse.com`), do you need to create your own boot disk as described here.

With Setup

Step by step

Here's how to create a boot disk:

1. Start setup directly from CD 1.
2. Select '`floppy`' and press ↵, next select '`Boot`' and again ↵.
3. Now you have to select a disk with a suitable kernel that, for example, supports your SCSI adapter if you have one. setup shows the essential part of the kernel descriptions. If you need further information, you can look it up in `\disks\readme.dos`. Remember the name of your your kernel. You will need it later. Now press ↵.
4. Now you're ready to create the boot disk. Insert the (DOS-formatted) disk into the 3.5" drive and select the disk you want to create.
   - Only the boot disk is needed ('`Root`' is not needed anymore for SuSE Linux.) Move the cursor onto '`Boot`' and press ↵.
   - setup requests you to confirm that you have inserted a disk. Press ↵ and the disk is written.
   - When this is finished, press ↵.
   - Now select '`Done`' to exit this screen and setup.

With rawrite

Alternatively you might want to use the (perhaps slower) DOS program rawrite.exe (CD 1, directory `\dosutils\rawrite`) to write the disk at the boot prompt.

The standard disk images are contained on CD 1 in the directory `/disks`; please read the file `README`. The image `bootdisk` or `scsi01` is the usual choice for the standard disk. All the actual kernels can be found in the directory `/suse/images` (without extensions). Also please read the `README` file there.

If you need the standard disk which is supplied with every SuSE Linux, proceed as follows: It is assumed that you are in the directory of the CD:

```
Q:> dosutils\rawrite\rawrite disks\eide01
```

```
Q:> dosutils\rawrite\rawrite disks\bootdisk
```

If you need a specific type of support, another disk image should be used instead of `bootdisk`. If problems arise, `k_i386` can be implemented as a fallback kernel.

2.4.4 Creating a Boot Disk with UNIX

Requirements

You need access to a Unix/Linux system with an accessible CD-ROM drive and a formatted disk.

This is how you create a boot disk:

1. If you need to format the disks first:

   ```
   earth: # fdformat /dev/fd0u1440
   ```

2. Mount the first CD (disk 1) (e. g., to `/cdrom`):

   ```
   earth: # mount -tiso9660 /dev/cdrom /cdrom
   ```

3. Change to the `disks` directory on CD:

   ```
   earth: # cd /cdrom/disks
   ```

4. Create the boot disk with

   ```
   earth: # dd if=/cdrom/disks/bootdisk of=/dev/fd0 bs=8k
   ```

   In the `README` file in the directory `disks` you can read about what features specific kernels have; these files can be read with **`more`** or **`less`**. .

5. If you need a different kernel, another disk image can be used in place of `bootdisk`. If problems arise, `k_i386` can be implemented as a fallback kernel.

Somewhat more complex is using a self-compiled kernel during installation. In this case, write the default image (`bootdisk`) to the disk and then overwrite the actual kernel (`linux`) with your own kernel (refer to Section 9.5 page 241):

```
earth: # dd if=/cdrom/disks/scsi01 of=/dev/fd0 bs=8k
earth: # mount -t msdos /dev/fd0 /mnt
earth: # cp /cdrom/suse/images/eide.ikr /mnt/linux
earth: # umount /mnt
```

2.4.5 Does Linux support my CD-ROM?

Almost every CD-ROM drive is now supported by Linux.

- Using ATAPI drives (those drives connected to an EIDE controller), there should be no problems at all.

- Using SCSI CD-ROM drives, it is only important whether the SCSI controller is supported by Linux (supported SCSI controllers are listed in the Appendix). If your SCSI controller is not supported and there is a HD connected to it, you will have a problem anyway.
- Lots of proprietary CD-ROM drives are supported under Linux (see Appendix). Here is the most likely place for problems to occur. If your drive is not mentioned, you could try choosing a similar type.
- In the meantime parallel port CD-ROM drives have become very popular. Unfortunately there is no standard, leading to unexpected trouble. SuSE Linux contains a number of alpha drivers for some devices. If none of them works, you have to install via DOS. Please keep in mind that you may access some of these devices only after they have been set up under DOS. You might need a warm reboot.

2.4.6 ATAPI CD-ROM Hangs While Reading

If your ATAPI CD-ROM is not recognized or it hangs while reading, this is most frequently due to incorrectly installed hardware. All devices must be connected to the EIDE controller in the correct order. The first device is master on the first controller; the second device is slave on the first controller. The third device should be master on the second controller, and so forth.

It often occurs that there is only a CD-ROM besides the first device. The CD-ROM drive is sometimes connected as master to the second controller (secondary IDE controller). This is wrong and can cause Linux not to know what to do with this "gap". You can try to fix this by passing the appropriate parameter to the kernel (`hdc=cdrom`) (see Section 10.3.2 page 247).

Sometimes one of the devices is just "mis-jumpered". This means it is jumpered as slave but is connected as master, or vice versa. When in doubt, just check your hardware settings and correct them where necessary.

In addition, there is a series of faulty EIDE chipsets, most of which have now been identified; there is a special kernel to handle such cases. (see the `README` in `/disks` of the installation CD-ROM). The kernel parameters needed are described in detail in Section 10.3.2 page 247 and Chapter 9 page 237.

Note

If booting does not work straightaway, try using the following kernel parameters. At the boot prompt (**boot:**), enter:
boot: `linux <relevant parameters>` (see below) *Note:* Don't forget to add the name of the kernel (`linux`) before the actual parameters!

`hd<x>=cdrom` – <x> stands for `a`, `b`, `c`, `d` etc. and is to be interpreted as follows:

- `a` – Master on the 1st IDE controller

- `b` – Slave on the 1st IDE controller
- `c` – Master on the 2nd IDE controller
- …

An example of <parameter to be entered>: `hdb=cdrom`

With this parameter you can specify the CD-ROM drive to the kernel, in case it cannot find it itself and if you have an ATAPI CD-ROM drive.

`ide`<**`x`**>**`=noautotune`** – <x> stands for `0`, `1`, `2`, `3` etc. and is to be interpreted as follows:

- `0` – 1st IDE controller
- `1` – 2nd IDE controller
- …

An example of <parameter to be entered>: `ide0=noautotune`

This parameter is often useful for (E)IDE hard disks.

Note

Further kernel parameters can be found in Section 10.3.2 page 247. In case of problems with SCSI systems or with connecting network cards, you should also look here.

2.4.7 Problems with CD-ROM Drives on Parallel Port

All available drivers are listed by linuxrc at the installation. Normally there are no peculiarities.

Unfortunately, lots of drives (e. g. Freecom) are not supported yet. It may be that you cannot use them although the manual claims that the type is identical. The manufacturer apparently has changed the internals without making these changes public.

Some of the devices need to be initialized by the DOS driver for making them available under Linux:

1. Boot DOS and load the CD-ROM driver.
2. Insert a Linux boot disk.
3. warm reboot the machine

If your drive is not supported you will have to install from a DOS partition, as before (see Section 2.3 page 33).

For current information on parallel port programming have a look at: `http://www.torque.net/linux-pp.html`.

2.4.8 Loadlin Doesn't Have Enough Memory to Load the Kernel

You don't have enough free memory below 640 KB. Try to remove drivers from your startup files or shift them to high memory.

If you use compressed drives under Windows 95/98 and shifting the driver to high memory doesn't work, you have to decompress those drives.

2.4.9 Loadlin Doesn't Start

If you encounter any problems using loadlin, you should start **`loadlin`** using the following options: `-v`, `-t` or `-d`.

It is best to write the debug information into a file, `debug.out`.

```
C:\> loadlin -d debug.out <other parameters>
```

You could send this file to the SuSE support. For <other parameters> you need to enter your system-specific values (see Section 4.9.1 page 118).

2.5 Partitioning Tips

Is Linux and its file systems completely new to you? Then you probably ask yourself questions like: How much space should I assign to Linux? What's the minimum I need? What's best for my needs? How should I divide up the available space?

Partition Types on a PC

Every hard disk contains a partition table has space for up to four entries. Each entry may be either a primary or an extended partition. Only *one* extended partition may be assigned.

Primary partitions are a continuous section of cylinders which are assigned to one operating system. Using primary partitions, you would only be able to create four partitions. More will not fit into the partition table.

An extended partition itself is also a continuous section of cylinders, but you can divide an extended partition into multiple *logical partitions* which, in turn, don't need a separate entry in the partition table. The extended partition is more or less a container for logical partitions.

If you need more than four partitions, you have to make one an extended partition in which you may assign logical partitions. The maximum is 15 partitions for SCSI systems and 63 for (E)IDE systems.

Linux doesn't care on what kind of partition it is installed. It can be primary or logical.

Time to Decide

Let’s start with the minimum SuSE Linux install: 180 MB. This only works if you use the machine for a simple purpose, e. g., you only work on a text console (no X Window System). If you want to take a look at X and start a few applications, you will need 500 MB. Both values include swap.

What is a reasonable size for an installation? 1 GB – in this world of several gigabyte hard disks, this is quite a modest requirement. And there is no upper limit.

What’s the best for your needs? That depends on what you want to do:

- Working in X and using applications such as **Applixware** and Netscape will require 1.2 GB.
- To create your own applications in X, you also need 1.2 GB.
- Both of the above items: 2 GB.
- To compile your own X servers, write your own CD’s together with the items mentioned above: 4 GB .
- To set up an Internet/FTP server: 700 MB minimum.

> **Tip**
>
> With the robustness of the Linux file system which has now been achieved, it is a very *good* idea, especially for novices, to follow the strategy suggested by YaST: a small partition at the beginning of the hard disk for `/boot` (at least 2 MB, for large drives, 1 cylinder), a partition for swap (64-128 MB), all the rest for `/`.

If you want to partition something, but also as little as possible, then the following simple rules are valid:

How should you divide up the hard disk? This simple question cannot be answered easily. Apply these guidelines:

- up to 500 MB: swap partition and a root (`/`)
- approx. 500 MB to 4 GB: small boot partition for the kernel and LILO at the *very beginning* of the hard disk (`/boot`, approx. 8-16 MB or 1 cylinder), a swap partition and the rest for the root partition `/`.
- more than 4 GB: boot (`/boot`, swap, root (250 MB), home (`/home` with approx. 200 MB per user) and the rest for applications (`/usr`); posssibly a further partition for `/opt` (see on the next page) and for `/var`.

> **Note**
>
> If you plan to start Linux directly from the hard disk, you will need a Linux partition below the “1024 cylinder limit” as a boot partition (you can read more on this in page 97). This doesn’t concern you if you are starting Linux from DOS/Windows with loadlin. Usually, the boot partition (`/boot`) will be the same as the root partition.

You should bear in mind that some programs (mostly commercial programs) install their data in`/opt`; if you are installing a number of these, either create a separate partition for `/opt` or enlarge the dimensions of the root partition accordingly. Specifically this involves the program packages or demo programs listed in table 2.1, calculated with size increases in mind (in the table mentioned there are also programs which are *not* included in SuSE Linux)

Just in case, either provide a partition for `/opt` or make the root partition big enough. Some examples are shown in table 2.1.

KDE	170 MB
GNOME	155 MB
htdig	5 MB
Fortify	2 MB
susehelp with htdig full text search	40 to 300 MB
Wabi	10 MB
Netscape	45 MB
Arcad	350 MB
Applixware	400 MB
Eagle	18 MB
StarOffice	260 MB
Cyberscheduler Software	30 MB
Cygnus Source-Navigator	20 MB
SNiFF+	45 MB
Insure++	45 MB
pep	18 MB
Oracle 8.1.7	900 MB
Sybase – Adaptive Server Enterprise	170 MB
virtuoso – OpenLink Virtuoso Lite Edition	55 MB

Table 2.1: Packages in directory `/opt`

2.6 Partitioning for Experts

In Section 2.5 page 43 and Section 3.3.11 page 64, details are given of the various ways to partition your system. This section should provide more detailed information for tailoring a system that best suits your needs. This section is mainly of interest for those who want an optimized system as far as security and performance are concerned and who are prepared to reinstall the complete system where necessary.

It is absolutely essential that you have extensive knowledge of the functions of a UNIX file system. You should be familiar with the topics mount point, physical, extended and logical partitions.

It should be mentioned here that there is *no* golden rule for all, but many rules for each situation. Don't worry, you will find concrete figures in this section to help you.

First, you need to gather the following information:

- What is the purpose of the machine (file server, compute server, stand-alone machine)?
- How many people are going to work with this machine (simultaneous logins)?
- How many hard disks are installed? How big are they and which kind (EIDE, SCSI or even RAID controllers)?

2.6.1 Size of Swap Partition

Quite often you will read:

“Swap should be at least as large as physical RAM”. This is a relic of times when 8 MB was regarded as a lot of RAM Memory.

Applications that need considerable memory have shifted these values up. Generally 64 MB of virtual swap should be sufficient. Do not be stingy. If you compile a kernel in X and want to have a look at the manual pages using Netscape and have an emacs running, you will already take up all 64 MB.

To be safe, opt for at least 128 MB of virtual memory. One thing you should never do is not assign swap space at all! Even on a machine with 256 MB RAM, there should be a swap partition.

Do you plan to run extensive simulations and need gigabytes of memory? When in doubt as to whether Linux suits your needs, please read Section 2.6.2 on the next page (Example: compute server).

2.6.2 Computer Used as Stand-alone Machine

The most common use for a Linux machine is as a stand-alone computer. To make decisions as easy as possible for you, we provide you with some concrete figures which you can use at home or at your company. Table 2.2 is an overview of size requirements for different Linux systems.

Installation	Disk space needed
minimum	180 MB to 400 MB
small	400 MB to 800 MB
medium	800 MB to 4 GB
large	4 GB to 8 GB

Table 2.2: Examples of disk space requirements for different installations

Example: stand-alone machine (small)

You have a 500 MB spare hard disk to hold Linux: use a 64 MB swap partition and the rest for / (root partition).

Example: stand-alone machine (average)

You have 1,2 GB available for Linux. A small boot partition `/boot` (5-10 MB or 1 cylinder), 128 MB for Swap, 800 MB for `/` and the rest for a separate `/home` partition.

Example: stand-alone machine (luxury)

If you have more than 1.2 GB available, there is no standard way to partition. Please read Section 2.6.3.

Using as a file server

Here, hard disk performance is *really* crucial. You should use SCSI devices if possible. Keep in mind the performance of the disk and the controller.

A file server is used to save data centrally. This data might be home directories, a database or other archives. The advantage of this is that administration of the data is simple.

If the file server will serve a huge net (20 users and more), optimizing hard disk access is essential.

Suppose you want to provide a file server for 25 users (their home directories). If the average user requires 100-150 MB for personal space, a 4 GB disk mounted under `home` will probably do.

If there are 50 such users, you will need an 8 GB disk. In this case, it would be better to split `home` into two 4 GB disks, as now they would then share the load (and access time!).

Using as a compute server

A compute server is generally a powerful machine that carries out extensive calculations over the net. Normally, such a machine is equipped with extensive main memory (512 RAM or greater). The only point where fast disks are needed is for the swap space. The same rule applies here: if you have a number of hard disks you can spread swap partitions across them.

2.6.3 Optimizations

The disks are normally the limiting factor. To avoid this bottleneck, there are two possibilities which should be used together:

- separate the load onto multiple disks
- use an optimized file system (e.g. `reiserfs`).
- equip your file server with enough memory (at least 256 MB)

Parallelizing multiple disks

This needs some further discussion. The total amount of time needed for transferring data can be separated into five factors:

- time elapsed until the request reaches the controller
- time elapsed until this request is send to the disk
- time elapsed until the hard disk manages to set its head
- time elapsed until the media has turned to the right sector
- time elapsed for transferring data

The first factor depends on the network connection and has to be regulated elsewhere. We do not cover this here. The second factor can be ignored, because it depends on the controller. The third factor is the vital part. The time is counted in milliseconds. Relative to the access time of main memory (measured in nanoseconds), this is a factor of one million! The fourth factor depends on the disk rotation speed. The fifth factor depends on the rotation speed, the number of heads and the actual position of the data (inside or outside).

For optimized performance, one should consider factor three. Here, the SCSI feature *disconnect* comes into play. Let’s look at what happens:

The controller sends the command (in this case to the hard disk) “Go to track x, sector y” to the device. Now the disk motor has to start up. If this is an intelligent disk (if it supports disconnect) and the driver itself is also able to do disconnect, the controller sends a disconnect and the disk separates itself from the SCSI bus. Now other SCSI devices can do work. After a time (depending on the strategy or load on the SCSI bus), a connection to the disk is re-established. Normally, the device has now reached the requested track.

On a multi-tasking, multi-user system like Linux, there are lots of optimizations that can be done here. Let’s look at an output of the command **`df`** (see output 2.6.1).

```
Filesystem     1k-blocks      Used Available Use% Mounted on
/dev/sda1        2974550    892663   1928065  32% /
/dev/sdb1        2974550   1948093    872635  69% /usr
/dev/sdc1        2457832   1967464    490368  80% /usr/lib
```

Output 2.6.1: Example of a **`df`** command output

So, what benefits does parallelizing bring us? Suppose we enter in `/usr/src`:

```
root@earth:/usr/src/ > tar xzf package.tgz -C /usr/lib
```

Here, package.tgz will be untarred into `/usr/lib/package`. To do so, the shell launches tar and gzip (located in `/bin` and thus on `/dev/sda`) then package.tgz in `/usr/src` is read (on `/dev/sdb`). At last, the extracted data is written to `/usr/lib` (on `/dev/sdc`). Using parallelizing, positioning as well as read/write of the disks’ internal buffers can be activated at the same time.

This is only one example; there are many more. If this example were a frequent processing requirement, then, as a rule of thumb, if there are many disks (with the same speed), /usr and /usr/lib should physically be placed on different disks. Here /usr/lib should have approximately 70% of the capacity of /usr. /, due to its access, should be placed on the disk containing /usr/lib.

From a certain number of SCSI disks onwards (4–5), one should consider buying a RAID controller. Thus, operations on the disks are not only quasi-parallel but parallel. Fault tolerance is one of its famous by-products.

Processing speed and size of main memory

The size of main memory is more important in Linux than the processor itself. One reason – if not the main reason – is Linux' ability to dynamically create buffers of hard disk data. Here, Linux uses lots of tricks, such as "read ahead" (getting sectors in advance) and "delayed write" (saving writes until there is a bundle to write). The latter is the reason why you should not switch off your Linux machine. Both items are the reason why Linux is so fast and why the memory seems to fill rapidly. Linux works with shared libraries, that is, lots of programs and applications share the same library. A library call has only to be allocated once. This also means that your memory fills up rather quickly. So if you do not know whether to purchase another main board or some more memory, we recommend you do the latter as it increases the speed of Linux.

```
             total   used    free   shared    buffers    cached
Mem:           255    246       9        0         23        44
-/+ buffers/cache:    178      76
Swap:          261      3     257
```

Output 2.6.2: Output of **free -m**

Output 2.6.2 shows that some 23 MB are being saved in buffers. If you want to access data that is already buffered, this data is almost immediately available.

3 Configuration with YaST

YaST (**`yast`**) is a very important part of SuSE Linux. YaST installs the system, manages the software (Installation and uninstallation) and provides system administration support.

This chapter outlines many important features of YaST, some of which are used during system installation.

> **Note**
>
> Your SuSE Linux 7.2 comes with the most recent version of YaST. Consequently it is possible that the screen shots included in this chapter may differ in detail from what is displayed on your monitor.

3.1 Basic Usage and Keyboard Layout

You can start YaST as user `'root'` from the Prompt simply by typing the command **`yast`**:

```
earth:~ # yast
```

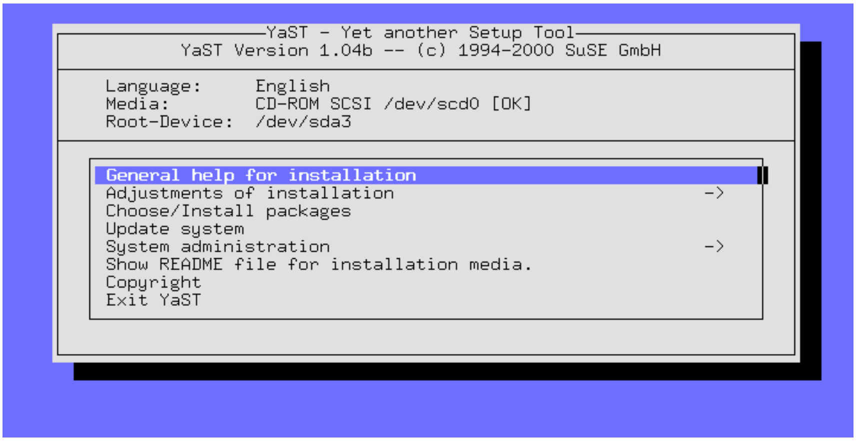
Navigate the menus and the screens in YaST with the arrow-keys and the tab-key (Tab). You can move around in lists with the cursor keys and with Page ↑ and Page ↓. Items in the lists can be selected with Space. Normally you can use the Enter (or ↵) key to confirm your choice. To return to a previous menu, use the Esc key. If you want to save a particular choice (such as the selection of a package you wish to install), use the F10 key which both saves the selection and returns to the previous menu.

Where you are prompted for a yes or no response, you can use the Tab key to switch back and forth between the two options. If you use YaST in color mode, the *active* selection is always highlighted in *blue*.

In rare cases, such as when you are accessing YaST remotely from a non Linux terminal, the function keys may not be usable. In these cases, the key combination, Ctrl +f <number> can be used as a substitute for F<number>. To simulate the F10 key, use the key combination Ctrl +f 0. There are no substitute mappings for the F11 and F12 function keys.

3.2 The YaST Main Menu

When you first start YaST, you will be in the “main menu” (Figure 3.1 on the next page).



```
                       YaST - Yet another Setup Tool
              YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH

     Language:     English
     Media:        CD-ROM SCSI /dev/scd0 [OK]
     Root-Device:  /dev/sda3

     General help for installation
     Adjustments of installation                              ->
     Choose/Install packages
     Update system
     System administration                                    ->
     Show README file for installation media.
     Copyright
     Exit YaST
```

Figure 3.1: The “main menu”

‘General help for installation’ As the title suggests, this menu option provides some general tips on installation.

‘Adjustments of installation ->’ This option will bring you a submenu for making adjustments to your installation. (see Section 3.3 on the facing page for more details).

‘Package management (Update, Installation, Requests)’ This menu option will bring you to the YaST package manager utility which is used to install software packages and remove them from the system (see Section 3.4 page 71).

‘Update system’ In case some packages need to be updated.

‘System administration ->’ Yet Another Self-explanatory Title. (see Section 3.6 page 78).

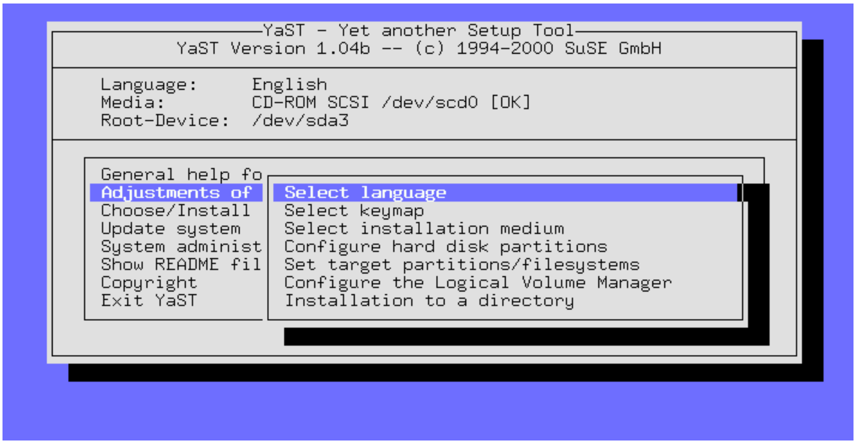
‘Show README file for installation media’ Important late-release information can be found here.

‘Copyright’ The lawyers have their say.

‘Exit YaST’ But not forever....

3.3 Adjustments of Installation

Adjustments to your installation can be made in one of the submenus of '`Adjustments of installation`' (Figure 3.2). With the Esc key you can return to the main menu.



```
─────────────────YaST - Yet another Setup Tool──────────────────
      YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH

   Language:     English
   Media:        CD-ROM SCSI /dev/scd0 [OK]
   Root-Device:  /dev/sda3

   General help fo
   Adjustments of   Select language
   Choose/Install   Select keymap
   Update system    Select installation medium
   System administ  Configure hard disk partitions
   Show README fil  Set target partitions/filesystems
   Copyright        Configure the Logical Volume Manager
   Exit YaST        Installation to a directory
```

Figure 3.2: The menu for '`Settings for installation`'

The actual installation can only be started once you have specified the target partitions! A number of items can be selected after the installation, which is useful if you want to update settings!

3.3.1 Selecting the Language

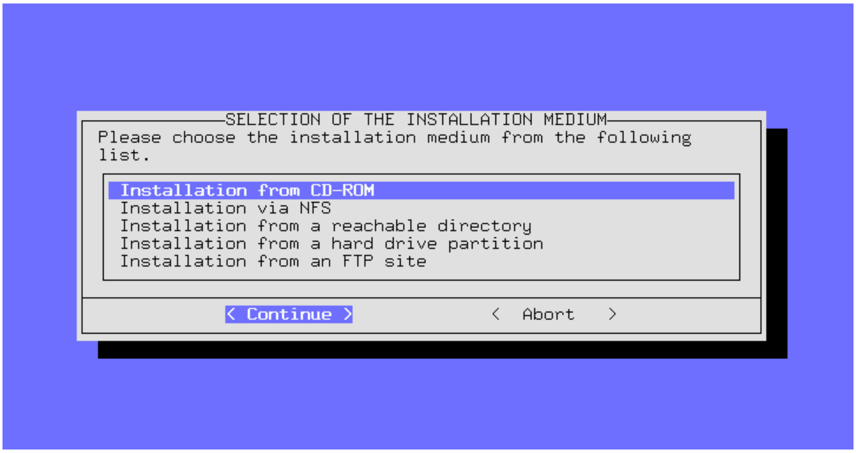
With '`Select language`', the language of the YaST mask can be changed. The value is stored in the variable **`DEFAULT_LANGUAGE`** in `/etc/rc.config`; see Section 12.9 page 308.

3.3.2 Selecting Keyboard Layout

With '`Select keymap`', the keyboard layout can be changed. The value is stored in the variable <KEYTABLE> in `/etc/rc.config` (see Section 13.6 page 318). Please choose a keytable according to the default language.

3.3.3 Installation Medium

Selecting the menu option '`Select Installation Medium`' will give you to select the source medium. From this menu (Figure 3.3 on the following page) you can select the medium from which you would like to install. This selection is made for installing the initial system and for adding packages to an existing system. YaST allows you, in another menu, the opportunity to select an installation medium, but this is for applying patches or updates to an existing system. This process is discussed later on in this chapter.



```
SELECTION OF THE INSTALLATION MEDIUM
Please choose the installation medium from the following
list.

  Installation from CD-ROM
  Installation via NFS
  Installation from a reachable directory
  Installation from a hard drive partition
  Installation from an FTP site

            < Continue >            <   Abort   >
```

Figure 3.3: Selection of the installation medium in YaST

In most cases you will want to install Linux directly from the CD-ROM so you will choose '`Installation from CD-ROM`'.

With the menu option '`Installation from hard disk-partition`' you have the option to install from a hard disk that contains the necessary source files. This option is helpful if your CD-ROM is not supported under Linux (see also Section 3.3.5).

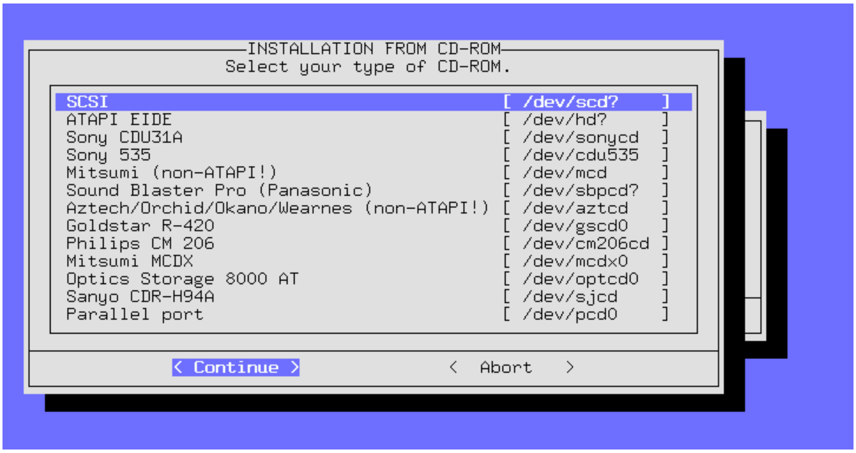
With the menu options '`Installation via NFS`' and '`Installation from a FTP-Server`', you can install Linux on a system that does not have a CD-ROM drive, but is connected by Ethernet to another computer that does have a CD-ROM drive. For additional information on these options, please refer to Section 3.3.6 on the facing page and Section 3.3.8 page 57.

3.3.4 Installation from CD-ROM

If you plan to install via CD-ROM, you must specify your CD-ROM drive (see Figure 3.4 on the facing page). If unsure, try '`ATAPI EIDE`' drives.

3.3.5 Installation from a Hard Disk Partition

If your CD-ROM drive is not directly supported, you can still install Linux on your system. To accomplish this, however, you will need to use a different installation medium.



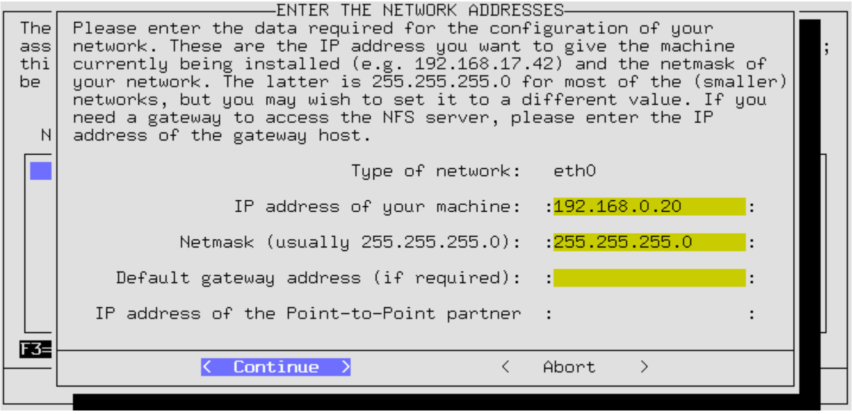
```
                 INSTALLATION FROM CD-ROM
               Select your type of CD-ROM.

 SCSI                                          [ /dev/scd?     ]
 ATAPI EIDE                                    [ /dev/hd?      ]
 Sony CDU31A                                   [ /dev/sonycd   ]
 Sony 535                                      [ /dev/cdu535   ]
 Mitsumi (non-ATAPI!)                          [ /dev/mcd      ]
 Sound Blaster Pro (Panasonic)                 [ /dev/sbpcd?   ]
 Aztech/Orchid/Okano/Wearnes (non-ATAPI!)      [ /dev/aztcd    ]
 Goldstar R-420                                [ /dev/gscd0    ]
 Philips CM 206                                [ /dev/cm206cd  ]
 Mitsumi MCDX                                  [ /dev/mcdx0    ]
 Optics Storage 8000 AT                        [ /dev/optcd0   ]
 Sanyo CDR-H94A                                [ /dev/sjcd     ]
 Parallel port                                 [ /dev/pcd0     ]

            < Continue >                 <  Abort   >
```

Figure 3.4: Selecting a CD-ROM drive

3.3.6 Installation via NFS

The network installation gives you the option of easily performing installations on multiple computers. This is especially useful when only one of the computers has a CD-ROM drive, from which the source files can be placed on an NFS reachable hard disk. Before attempting to install via NFS, you should be familiar with how to configure NFS servers.

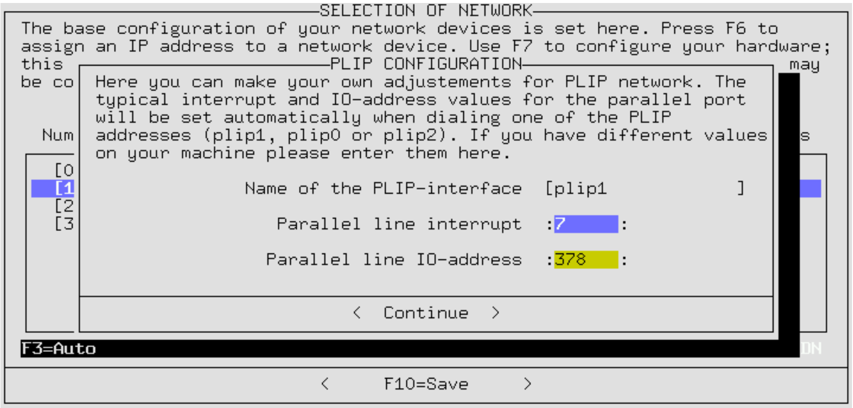
It is also possible to use this method to install Linux onto notebooks which have a functional PCMCIA network card.



```
                   ENTER THE NETWORK ADDRESSES
 Please enter the data required for the configuration of your
 network. These are the IP address you want to give the machine
 currently being installed (e.g. 192.168.17.42) and the netmask of
 your network. The latter is 255.255.255.0 for most of the (smaller)
 networks, but you may wish to set it to a different value. If you
 need a gateway to access the NFS server, please enter the IP
 address of the gateway host.

                        Type of network:    eth0

                  IP address of your machine:   :192.168.0.20        :

             Netmask (usually 255.255.255.0):   :255.255.255.0       :

       Default gateway address (if required):   :                    :

     IP address of the Point-to-Point partner   :                    :

               < Continue >               <  Abort   >
```

Figure 3.5: Entering network data for an NFS installation

Installation via NFS is not only available to computers that are connected together via ethernet. The NFS installation can also be accomplished with computers that are “networked” via their parallel port. This feature is of particular interest for users who want to install Linux onto laptops and notebooks. If you are planning to attempt such an installation, naturally you must remember, when booting, to select a kernel that contains PLIP support. Additionally, in this case, you will need to provide additional entries so that the PLIP interface can be correctly configured:



```
──────────────────────SELECTION OF NETWORK──────────────────────
The base configuration of your network devices is set here. Press F6 to
assign an IP address to a network device. Use F7 to configure your hardware;
this ──────────────────PLIP CONFIGURATION──────────────────   may
be co  Here you can make your own adjustements for PLIP network. The
       typical interrupt and IO-address values for the parallel port
       will be set automatically when dialing one of the PLIP
  Num  addresses (plip1, plip0 or plip2). If you have different values   s
       on your machine please enter them here.
  [0
  [1                 Name of the PLIP-interface  [plip1           ]
  [2
  [3                 Parallel line interrupt     :7      :

                     Parallel line IO-address    :378    :

                                < Continue >
F3=Auto
                          <      F10=Save      >
```

Figure 3.6: PLIP configuration

If you are directly connected to the NFS server through the parallel port, then the IP address of the PLIP partners is the same, as is shown in the following examples of the NFS servers. The PLIP interface in most cases will be `plip1`. By “interface” we are referring to the network interface that can be seen and configured with the command “ifconfig”. The hardware parameters will differ only in rare cases.

You should be aware that, by using the parallel port as a PLIP interface, printers will not function on this interface! In many cases a printer connected to such a port will lead to a permanent reset as soon as the driver kicks in.

In the following screen you must enter the IP address of the NFS server as well as the directory in which the source files are located. Of course, the NFS server must have exported this directory so as to make it available to the computer you are installing!

3.3.7 Installation from a Reachable Directory

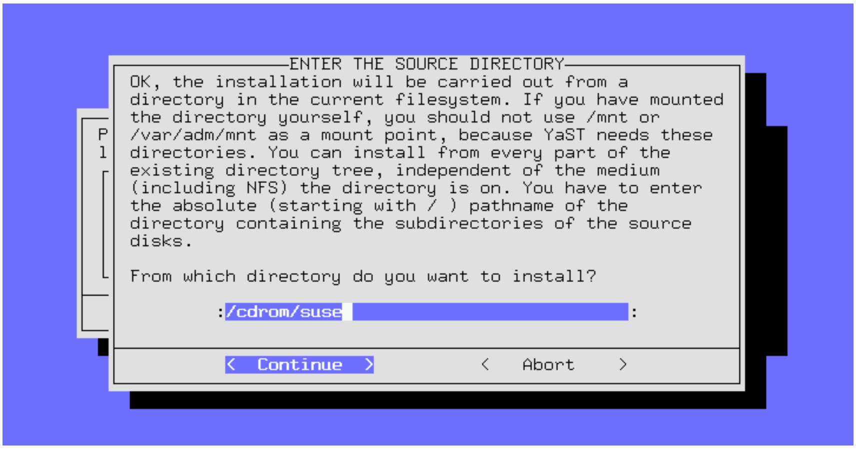
With this option it is possible to install addtional software later if you already have your SuSE Linux running. This is a useful option when you are unable to install using a CD-ROM, due to the unavailability of drivers.

To be able to access this drive from YaST, first start YaST. Once this is accomplished, you need to switch to another console. For example, with the Alt +F2 key combination you will be presented with the second virtual console. Login as `‘root’`.

From here you must mount the CD-ROM drive *manually* onto an available mount point in your directory tree, for example with the command

```
earth: # mount -tiso9660 /dev/cdrom /cdrom
```

Now you can follow through with a normal installation. To accomplish this, in the following screen (Figure 3.7) you must enter the directory in which the source files are located. This should be the directory onto which the CD has been mounted, followed by the directory `suse`.



```
ENTER THE SOURCE DIRECTORY
OK, the installation will be carried out from a
directory in the current filesystem. If you have mounted
the directory yourself, you should not use /mnt or
/var/adm/mnt as a mount point, because YaST needs these
directories. You can install from every part of the
existing directory tree, independent of the medium
(including NFS) the directory is on. You have to enter
the absolute (starting with / ) pathname of the
directory containing the subdirectories of the source
disks.

From which directory do you want to install?

        :/cdrom/suse                          :
```

Figure 3.7: Entering the source directory

If for example you had mounted the drive (or the hard disk partition) onto the directory `/cdrom`, then you would use the field as depicted in Figure 3.7:

```
/cdrom/suse
```

as the directory for the source files.

3.3.8 Installation via FTP

In a similar way to the NFS install, it is possible to install SuSE Linux onto a computer that has no CD-ROM at all, via FTP. This will work, but only if the basic network configurations have been set up correctly.

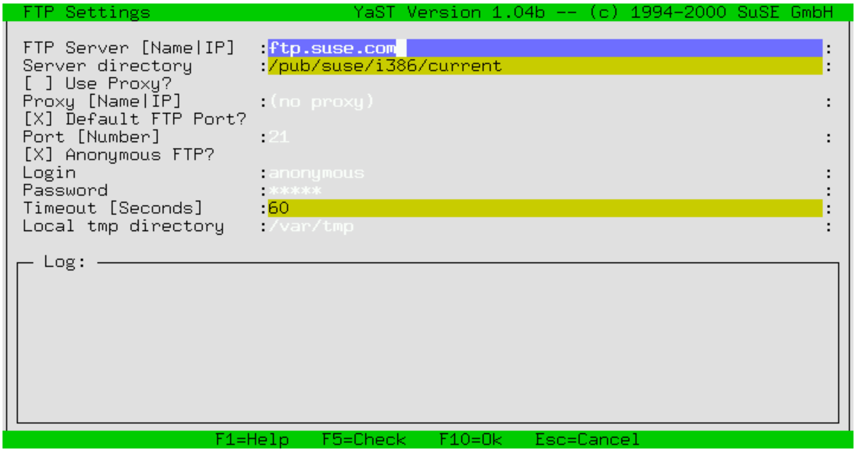
'`FTP Server [Name|IP]`' The name or the IP address of the FTP server.

'`Server Directory`' The location on the FTP server where the `suse` directory is located.

'`[ ] Use Proxy?`' Only select this if you are sure that you must use an FTP proxy server. Normally this is *not* necessary.

'`Proxy [Name|IP]`' This is only necessary if you have opted to use the proxy server.

'`[X] Default FTP Port?`' This should be selected by default.



```
 FTP Settings                                YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH

 FTP Server [Name|IP]    :ftp.suse.com                                                     :
 Server directory        :/pub/suse/i386/current                                           :
 [ ] Use Proxy?
 Proxy [Name|IP]         :(no proxy)                                                       :
 [X] Default FTP Port?
 Port [Number]           :21                                                               :
 [X] Anonymous FTP?
 Login                   :anonymous                                                        :
 Password                :*****                                                            :
 Timeout [Seconds]       :60                                                               :
 Local tmp directory     :/var/tmp                                                         :

 ┌ Log: ─────────────────────────────────────────────────────────────────────────────────┐



 └───────────────────────────────────────────────────────────────────────────────────────┘
                     F1=Help    F5=Check    F10=Ok    Esc=Cancel
```

Figure 3.8: Entries for the FTP installation

`'Port [Number]'` The default value should be `21`.

`'[X] Anonymous FTP?'` This should always be selected if you want to access a public FTP server.

`'Login'` In the case where you have *not* selected the anonymous FTP option above, here you can insert a username and in the following field

`'Password'` the password.

`'Timeout [Seconds]'` `60` is the suggested value.

`'Local /tmp directory'` The local directory that is available for temporary storage.

3.3.9 Partitioning Your Hard Disk

The most critical point of installing a new operating system is partitioning the hard disk. Generally, every operating system uses at least one partition.

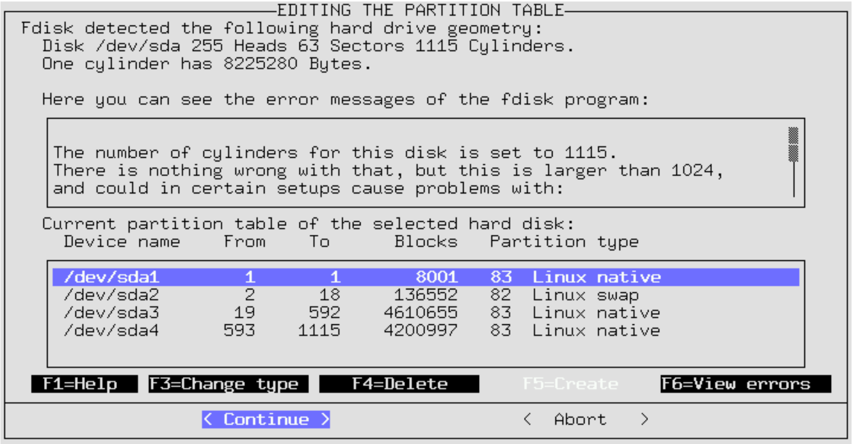
Tip

With Linux it is also possible to install the system on an existing MS-DOS file system, but you should use this option only to "browse through". The performance is substantially less than when installing Linux in its own Linux-specific partitions and the system is not as secure since there are no file check systems available for MS-DOS and Linux could be influenced while DOS is running.

With Linux systems, you will generally want at least three partitions, one (V)FAT partition for the bootloader (`/boot`), one for files and programs and one for memory data swapping which is performed by the running system. You might consider creating multiple partitions for files and programs. The number and sizes of partitions in a Linux system is a question of personal taste and philosophy (see Section 2.5 page 43 and Section 2.6 page 45); a detailed example of partitioning can be found in Section 3.3.11 page 64.

In addition to the partition (or partitions) that you create for Linux, you should also create a swap partition to increase the size of the computer's virtual memory (see Section 2.6.1 page 46) . It is possible to use a swap file instead of a swap partition. For performance reasons, however, this is not recommended. Every time your system accesses this file, it must do it through the file system. Systems with limited memory should always use a swap partition.

If you have more than one hard disk on your system, you will be prompted to select the drive that you wish to partition. Once you have selected the drive, you will see a menu which shows you the current status of the partitions on your hard disk (Figure 3.9).



```
                      EDITING THE PARTITION TABLE
 Fdisk detected the following hard drive geometry:
   Disk /dev/sda 255 Heads 63 Sectors 1115 Cylinders.
   One cylinder has 8225280 Bytes.

   Here you can see the error messages of the fdisk program:

    The number of cylinders for this disk is set to 1115.
    There is nothing wrong with that, but this is larger than 1024,
    and could in certain setups cause problems with:

   Current partition table of the selected hard disk:
     Device name       From     To        Blocks    Partition type

     /dev/sda1            1      1          8001    83  Linux native
     /dev/sda2            2     18        136552    82  Linux swap
     /dev/sda3           19    592       4610655    83  Linux native
     /dev/sda4          593   1115       4200997    83  Linux native
```

Figure 3.9: Partitioning with YaST

The arrow keys ↑ and ↓ can be used to navigate through the list of existing partitions. To change the type of file system associated with a partition, for example from DOS to ext2, use the F3 key. To delete existing partitions, use the F4 key. To create new partitions, use the F5 key.

Changing the type of the partitions (using the F3-key) is necessary, if you're creating a swap-partition. Sometimes it's not necessary to change the partitioning at all, especially when you decided to use the partition-layout from the operating system installed before. In this case just select the respective partition and press the F3-key.

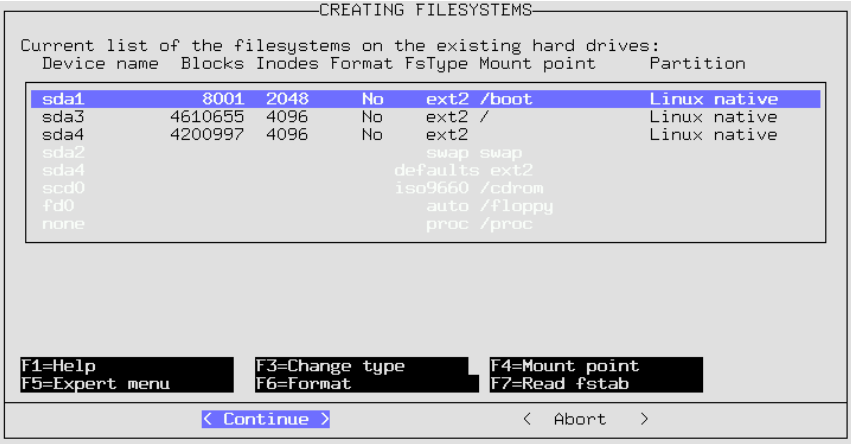
Be aware that Linux has no preferences regarding on what type of partition it gets installed:

It is equally acceptable to install Linux on either a *primary* partition or a *logical* partition within an extended partition. The logical partition is a workaround to limitations of modern day hardware. It was introduced to overcome the fact that the partition table is only large enough for four entries. If you are interested in having more than four partitions on your hard disk, you will need to create one *extended* partition. Once the extended partition is created, you will be able to create multiple *logical* partitions within this one large *extended* partition.

> **Note**
>
> Extended as well as logical partitions are known to DOS as *extended DOS partition* or *logical drive*, although this concept applies to any operating system.

3.3.10 Setting Target Partitions/File systems

After you have completed partitioning your hard disk, you must next specify how the partitions will fit into the Linux directory tree. Select the menu option '`Set target partitions/filesystems`'.



```
Current list of the filesystems on the existing hard drives:
  Device name  Blocks Inodes Format FsType Mount point     Partition

  sda1            8001  2048      No   ext2 /boot           Linux native
  sda3         4610655  4096      No   ext2 /               Linux native
  sda4         4200997  4096      No   ext2                 Linux native
  sda2                                 swap swap
  sda4                             defaults ext2
  scd0                              iso9660 /cdrom
  fd0                                  auto /floppy
  none                                 proc /proc
```

Figure 3.10: Assigning file systems

In Figure 3.10), you can see an example of partitioning a hard disk. For each individual partition you can select if and how it should be formatted as well as at what location in your directory tree it should be "mounted".

You must specify one partition as the "root" partition! This partition is necessary because it is the starting point of the entire directory structure, and yes, this *is* analogous to the function that roots provide for trees! Make sure you assign Mountpoint '`/`' as the mount point for your "root" partition.

The menu at the bottom of your screen defines the different function keys that can be used to manipulate the file system. As a rule you only have this functionality available to you on Linux partitions. If you highlight partitions that are formatted for other file systems, the function keys will not be available for use, except for the F3 key which is used to select the mount point.

Setting the file system type

With F3 you can choose between the two file systems, `ext2` and `reiserfs`.

‘`ext2`’ – The `ext2` file system (*second extended file system*) has been established for many years as the standard file system.

‘`reiserfs`’ – *ReiserFS* is a file system for the next generation; although this file system is still very new, it is already widely used – among other things, on a number of our internal systems. You really should read the notes in `/usr/share/doc/packages/reiserfs/README`.

In close cooperation with the developer group surrounding Hans Reiser and Chris Mason, SuSE has extended the high-performance file system to include a “journaling file system”. Through journaling, it is guaranteed that even for large servers, a file system check can be carried out within a few seconds.

> **Caution**
>
> Please do *not* use ReiserFS together with RAID 5 software. For hardware RAID solutions there are no such limitations. At present a ReiserFS partition must be at least 34 MB in size.

The following points should be noted if you want to use ReiserFS for an NFS server:

- knfsd generally works, but problems may arise with the extensive renaming of special files (“device files”, “named pipes”) via NFS across different directories.
- unfsd functions reliably if the file system in question is exported with the option `no_cross_mounts` or if unfsd is started with `--no-cross-mounts`.
- Exporting entire file system hierarchies (possible with unfsd) does not function reliably with ReiserFS.

The utilities for ReiserFS are in the package `reiserfs`, series a. Current information can be found at:

`http://www.namesys.com/`

Mountpoint

The F4 key can be used to select the location where the selected partition should be mounted on the directory tree.

Just a reminder: You *must* have *one* partition assigned as the root partition (`/`).

If you already have DOS/Windows partitions on your drive, you can also create mount points for them. For example, you could create a mount point of `/dosc` for the first DOS partition and a mount point of `/dosd` for the second.

> **Caution**
>
> Make sure that you assign absolute path names (starting with (`/`)) for all of your mount points and that you create one (and only one) unique mount point for each partition! In addition, you should never create separate partitions for the `/etc`, `/bin`, `/sbin`, `/lib` and `/dev` directories. These directories, which contain important commands, libraries and configuration files that are necessary for mounting the rest of the file system, must reside on the root partition!

An important note regarding DOS/Windows partitions! These partitions can be connected to the directory tree as one of three different types.

- As a “normal” **`DOS`** partition (= `msdos`), with all of the normal limitations of a DOS file system,
- as a **`FAT-Win95`** (= `vfat`) in which long filenames are possible
- in the third case, as the **`UMSDOS`** partition, which permits the use of long filenames in a “normal” DOS partition. This type is only required, however, if you want to store Linux files on a DOS partition; this is not very efficient, though, if you only need the DOS partition to store data. Whenever possible, you should *avoid* **`UMSDOS`**!

Expert Menu for Fine Tuning of the File system Inode Density

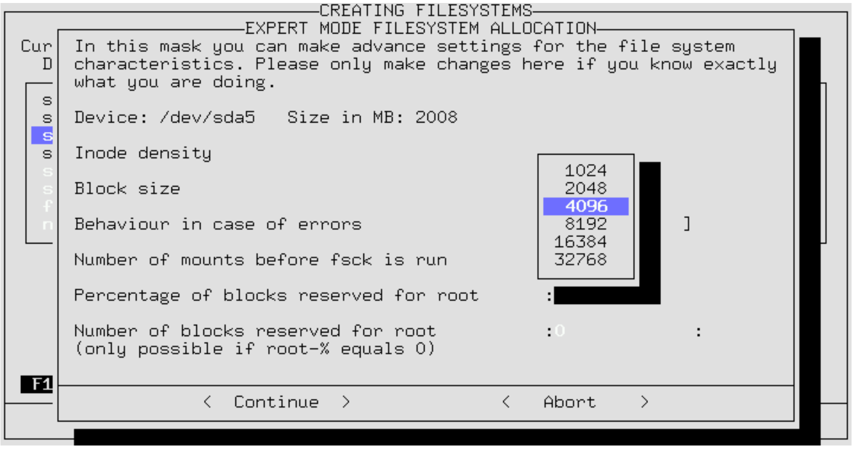
Here in the Expert menu fine tuning can be performed.

The Inode density is used to define the anticipated average file size (per inode) on a particular partition. If, for example, you choose a density of 4096 Bytes per inode, this means that you expect the average file size to be about 4096 Bytes (4 KB).

When a file system is created on a partition, the system computes the number of inodes it will need, using the inode density number and the size of the partition. The number of inodes represents the number of files that can be created on a given partition.

In the example above, if the partition’s size is 4 megabytes, the partition would have less than 1000 inodes. If this partition is filled with many files that are all approximately 1 KB, then only one fourth of the partition could be used before all of the inodes were exhausted. No more files could be added to the partition.

The inode value of 4 KB per file has proven to be a good default value (an excessive number of inodes would result in the unnecessary consumption of disk space, since inodes need disk space too, of course). If a particular partition is



```
                              CREATING FILESYSTEMS
                        EXPERT MODE FILESYSTEM ALLOCATION
 Cur  In this mask you can make advance settings for the file system
   D  characteristics. Please only make changes here if you know exactly
      what you are doing.
   s
   s  Device: /dev/sda5   Size in MB: 2008
   s
   s  Inode density
                                                  1024
      Block size                                  2048
                                                  4096
      Behaviour in case of errors                 8192             ]
                                                  16384
      Number of mounts before fsck is run         32768

      Percentage of blocks reserved for root     :

      Number of blocks reserved for root         :0               :
      (only possible if root-% equals 0)
 F1
                  < Continue >               < Abort     >
```

Figure 3.11: Expert menu for setting file systems

to be used to perform spooling (necessary for the NEWS service, for example) a density of 2048 Bytes per inode is recommended, because news articles are typically very small files.

Deciding on the *block size* has a direct influence on how long a file system check lasts: The duration of the file system check is quadratically inversely proportional to the block size - i. e. the smaller block, the more blocks and therefore longer check time. The suggested block size is `1024` (= 1 KB) for partitions up to 1 GB, stepping to `4096` (= 4 KB) for 4 GB partitions.

Specify the area reserved for `'root'`. For `'root'` you should always reserve an area if partitions are to be used both by system programs and by regular users. A separate `/home` partition means that space does not have to be set aside for `'root'`.

The *behavior on errors* can be specified. If an error occurs in the file system, operation can proceed as normal (`continue`), the partition can be re-mounted as read-only (`remount-ro`), or the complete system can be halted immediately (`panic`).

Caution

The option you choose depends on what for your system is used. The default `continue` should remain reserved for experienced system administrators.

Formatting the Partitions

With the F6 key you can select if and how the partitions should be formatted. With brand new disks it is not necessary to check for bad sectors while formatting, but to be on the safe side you can choose to do so. Selecting the formatting and check option will take quite a bit longer than just formatting a partition.

Reading the `fstab` File

The file `fstab` tells Linux what file systems to mount when it loads. If you have Linux already installed on your hard disk, you can press the F7 key to read an existing `fstab` file. The mount points listed in the `fstab` file will be automatically included for existing non-swap partitions on your hard disk. Entries in the `fstab` file for other types of file systems (`swap`, `proc`, `nfs-mount`, CD-ROM entries, etc.) will appear grayed out and cannot be changed. They will remain unchanged when you save the `fstab` file.

This feature is used when you are updating your entire base system (see Section 11.1.3 page 280) . In such cases YaST needs to know how your file system and partitions are divided. It is possible to have more than one Linux system installed on your computer.

3.3.11 Configuring Your Hard Disk Manually

Configuring Partitions

What's it all about?

Background information on the various partition types can be found in a document by Andries Brouwer (`http://www.win.tue.nl/~aeb/partitions/`).

Step by step...

This is how to proceed when setting up your partitions:

1. YaST presents you with a screen that is divided into several parts (see figure 3.9 page 59):
   - At the top, you should see your hard disk's parameters.
   - The second part shows warnings and error messages from the program fdisk. You can take a closer look by pressing F6.
   - At the bottom you can see the partitions fdisk has found. Here, you should see the partitions of your former operating system and the swap partition, if you have already assigned one.
2. If you want to remove existing partitions to use the space for new partitions, you should do that now.

   Highlight the partition you want to remove. Remember ↓ and ↑ lets you scroll through the list. Make sure the correct partition is highlighted. Then press F4 and confirm by pressing ↵.

> **Caution**
>
> If you remove a partition, all data on this partition will be lost.

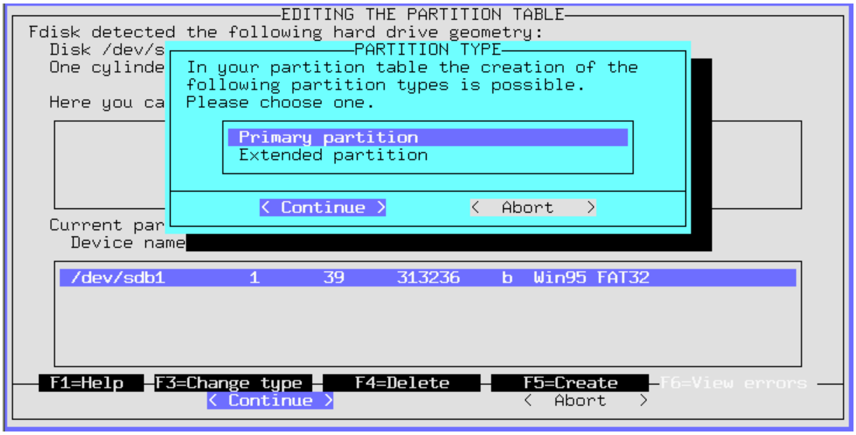
If you want to remove multiple partitions, you should do that now.

3. If you want to use existing partitions of other systems for Linux, you may do so simply by altering the partition type.

> **Caution**
>
> If you alter the partition type, other operating systems such as DOS and Windows may not be able to access data contained on them!

Now highlight the partition which you want to change, using ↓ und ↑. If you are sure you have marked the “correct” partition, just press F3. Now you are presented with a dialog box where you select the partition type. Select either *normal* or *swap* and confirm by pressing ⏎.

4. If you want to assign new partitions (this is the normal case), you should create them sequentially. Just press F5. If this doesn't have any effect, it means that the disk is already full with partitions. Then you may have to delete partitions. (See above).



```
─────────────────────────EDITING THE PARTITION TABLE──────────────────────────
 Fdisk detected the following hard drive geometry:
   Disk /dev/s ──────────────PARTITION TYPE──────────────
   One cylinde  In your partition table the creation of the
                following partition types is possible.
   Here you ca  Please choose one.

                    Primary partition
                    Extended partition

                      < Continue >          <  Abort   >
   Current par
     Device name

    /dev/sdb1        1       39      313236      b   Win95 FAT32

   F1=Help  F3=Change type    F4=Delete     F5=Create   F6=View errors
                  < Continue >                  <  Abort   >
```

Figure 3.12: YaST – Assigning Partitions

A dialog box pops up letting you select the type of partition. (figure 3.12)

You may select between either ‘`primary partition`’, ‘`extended partition`’ or ‘`logical drive`’. Confirm by pressing ⏎. Remember: you may assign up to a maximum of four *primary* partitions. If you need more, you need to assign at least one as an *extended* partition. Within this *extended* partition, you may assign *logical partitions*.

When you have assigned a primary or logical partition, you must enter the device name. YaST displays a list of the available device names. Normally, you should just select the uppermost entry, e. g., ‘`/dev/hda2`’ and press ⏎.

The next step lets you set the size of the partition. YaST recommends starting at the first cylinder of the first free partition. Normally, you should confirm this default. To set up the size, you may select one of three alternatives: the number of the last cylinder (e. g., `976`), the number of cylinders of the partition (e. g., `+66`) or the size in megabytes (e. g., `+100M`). Pressing ↵ should now move you to '`Continue`'. Here, just press ↵.

Now you should see your freshly assigned partition in the lower part of the screen. If you are not satisfied, you may remove it by pressing F4.

5. One of your partitions needs to be a swap partition. If there isn't already a swap partition, use ↓ and ↑ to select a proper one. Press F3 and select '`Linux Swap Partition`' and confirm by pressing ↵.

6. Is everything to your satisfaction? The drive should be completely populated with partitions. Did you remember the swap partition? Now move the cursor to '`Continue`' and ↵ then moves you to the '`Assigning mount points`' menu.

Under certain circumstances, YaST checks your swap partition to make sure no errors occurred.

3.3.12 File systems and Mount Points

What's it all about?

In the previous section all partitions were written to the partition table. In this step you will now provide additional information for the Linux partitions which have just been configured, and also for any DOS partitions, if they exist.

You have created all necessary partitions and these partitions have been added to the partition table, but they are still not written to disk. There are some vital parameters still missing which will be added in this step.

Summary

This information is partially written to the file system table, `/etc/fstab`, which contains all the relevant information on the file systems you want mounted. Each entry includes the device, its position in the directory tree and the type of file system.[1]

Furthermore, YaST needs this data to create Linux file systems on the formatted drives. The swap partition is not touched in this step as it has been already created in Section 3.3.11 page 64.

In contrast to the partition table entries (see Section 3.3.11 page 64), the `/etc/fstab` entries are Linux-specific and have no influence on other operating systems.

Also worthy of explanation:

[1] As well as information for dump and fsck; see manpage for **`fstab`** (**`man 5 fstab`**).

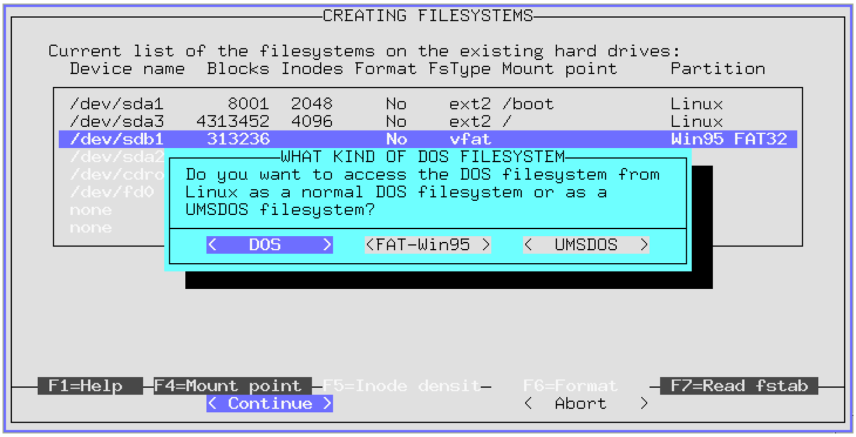
- In Linux, all file systems are linked to one "directory tree". You must assign each file system a branch on this tree. This is called its mount point. Even DOS file systems can be linked into the Linux directory tree.

- Hard disk space for file systems is administered in Linux by using Inode. An inode is just a small file that points to the corresponding data for a given file. The number of inodes is assigned while creating a file system. If you want to create lots of small files, you will need many inodes, which require some space of their own as well. For file systems containing many large files, fewer inodes are necessary. For more detailed information on this, see Section 3.3.10 page 62.

Step by step

You are now in '`Set target partitions / file systems`' (see figure 3.10 page 60). To determine the file systems of your new partitions, proceed as follows:

1. First some information about what is to be done:
   - For each of your DOS file systems (in DOS partitions), you *may* create a mount point.
   - For each of your new Linux partitions:
     - you *must* select a mount point
     - you *may* select a file system '`F3=choose type`' (`ext2` or `reiserfs`).
     - you *may* do some fine tuning, with '`F5=Expert menu`'
     - you *may* change the suggested formatting mode '`F6=Formatting`' (*with* or *without* a check).
   - The function '`reading fstab`' is normally not needed.



```
                         CREATING FILESYSTEMS

  Current list of the filesystems on the existing hard drives:
    Device name  Blocks Inodes Format FsType Mount point     Partition

    /dev/sda1       8001  2048     No     ext2 /boot          Linux
    /dev/sda3    4313452  4096     No     ext2 /              Linux
    /dev/sdb1     313236           No     vfat                Win95 FAT32
    /dev/sda2 ┌──────WHAT KIND OF DOS FILESYSTEM──────────────────┐
    /dev/cdro │ Do you want to access the DOS filesystem from     │
    /dev/fd0  │ Linux as a normal DOS filesystem or as a          │
    none      │ UMSDOS filesystem?                                │
    none      ├───────────────────────────────────────────────────┤
              │    <   DOS   >   <FAT-Win95 >   <   UMSDOS   >    │
              └───────────────────────────────────────────────────┘

   F1=Help   F4=Mount point  F5=Inode densit   F6=Format   F7=Read fstab
                < Continue >                      <  Abort    >
```

Figure 3.13: YaST – mount DOS partitions

2. You can set the type of the Linux file system with F3. Currently the tried and tested `ext2` or the new `reiserfs`, with advanced options, are available.

> **Caution**
>
> ReiserFS can *not* be used in conjunction with RAID 1 or RAID 5 software. For hardware RAID solutions there are no limitations. `reiserfs` is very new in comparison to `ext2`. Please bear this in mind if you decide on `reiserfs`. There is more information about ReiserFS on page 61.

3. If you want to access a DOS partition in Linux, select it and press F4. Now a window pops up and you can enter a directory. In this directory, you will mount your DOS file system. For example, enter `/dosc` (don't forget the slash!). Confirm this action by pressing ↩.

   If you have selected a DOS partition, another window appears (figure 3.13 on the page before). Here, you can choose which method to use to access DOS in Linux (see Section 3.3.10 page 61). Generally you will want to use your DOS file system independently and only occasionally want to access the DOS partition from Linux, for example, to transfer files. For this purpose, choose '`DOS`' or '`FAT-Win95`' (= `vfat`). UNIX file attributes and long file names, as provided by the UMSDOS file system, are *only* needed if you want to install SuSE Linux on a DOS partition – do this at your peril!

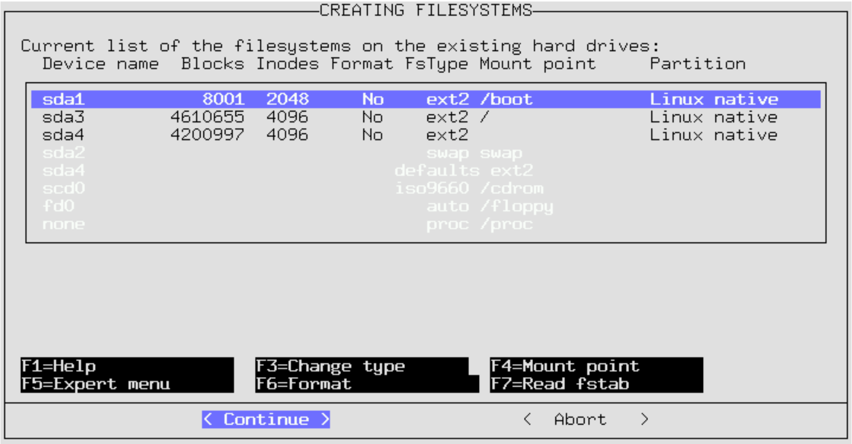
4. Next, enter the mount points for your Linux partitions. Just select your Linux partitions one after the other and press F4. A window will pop up letting you decide where to mount this partition on your file system tree. Enter the complete path beginning with the "slash" (/).

   Which mount points you choose for your partitions depends on how you plan to organize them. In any case, you *definitely* need a root directory `/`, which is the "root" for the whole file system. All other file systems and their mount points more or less depend on your preferences. You should create a mount point for each Linux file system. You will not be able to access unmounted file systems.

> **Caution**
>
> Directories which are needed at boot time must be located directly under `/`. At this stage, the separate branches are not yet put together (mounted). For this reason you may not assign `/bin`, `/dev`, `/lib`, `/etc`, and `/sbin` to other partitions.

5. You should ignore the '`Expert menu`' if you are just starting out with Linux; as a rule defaults are set automatically to guarantee a secure operation. Only change the suggested values if you really know what you are doing (see page 62)!

6. You must now decide whether (and how) you want to format your Linux partitions. Since you just created those partitions in Section 3.3.11 page 64, they must now be formatted. If you own a fairly new hard disk, do '`normal formatting`'. If your storage media is not that new, it is better to choose '`Formatting and checking`'.



```
                        CREATING FILESYSTEMS
 Current list of the filesystems on the existing hard drives:
   Device name   Blocks Inodes Format FsType Mount point     Partition

   sda1            8001  2048    No     ext2 /boot           Linux native
   sda3         4610655  4096    No     ext2 /               Linux native
   sda4         4200997  4096    No     ext2                 Linux native
   sda2                                 swap swap
   sda4                            defaults ext2
   scd0                             iso9660 /cdrom
   fd0                                  auto /floppy
   none                                 proc /proc

 F1=Help              F3=Change type       F4=Mount point
 F5=Expert menu       F6=Format            F7=Read fstab

                < Continue >                  <   Abort   >
```

Figure 3.14: YaST – Mount points

Select the Linux partitions one after another and press (F6) to select the appropriate formatting. At the end the menu should resemble figure 3.14 (it is possible you have *not* specified an NFS directory!).

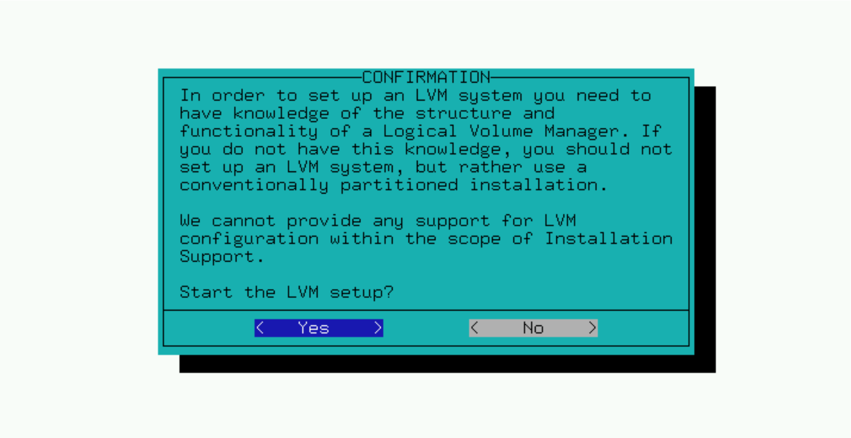
7. After selecting '`Continue`', a dialog box pops up asking you whether you are sure. Then YaST formats your partitions.

3.3.13 Configuring the Logical Volume Manager

The Logical Volume Manager (LVM) enables you to distribute the hard disk space over the various file systems in a flexible manner. LVM was developed to alleviate the relative complicated process of modifying partitions on a running system: It makes a virtual "pool" (Volume Group – or VG for short) of memory space available, from which logical volumes (LV) can be generated as needed. The operating system will then access these instead of the physical partitions.

Features:

- Multiple hard disks/partitions can be combined into one large logical partition/.
- If a LV (e.g. `/usr`) gets full, you can expand it given the appropriate configuration.
- With the LVM, you can even append hard disks or LVs on a running system. "Hot–swapable" hardware is, however, required for such intervention procedures.



```
CONFIRMATION
In order to set up an LVM system you need to
have knowledge of the structure and
functionality of a Logical Volume Manager. If
you do not have this knowledge, you should not
set up an LVM system, but rather use a
conventionally partitioned installation.

We cannot provide any support for LVM
configuration within the scope of Installation
Support.

Start the LVM setup?
```

Figure 3.15: YaST – Logical Volume Manager

Using LVM is even worthwhile for home PCs used extensively or even for small servers. If you have a growing data stock such as in the case of databases, MP3 archives or user directories etc., the Logical Volume Manager is just the right tool for you. With this, you can have file systems, for example, which are larger than a physical hard disk. Another advantage of the LVM is this: You can create up to 256 logical volumes. Please note that working with the LVM, however, is quite different from working with conventional partitions.

Further information on configuring the “Logical Volume Manager” (LVM) can be found in the official LVM Howto:

http://www.sistina.com/lvm/Pages/howto.html

3.3.14 Installation to a Directory

With YaST you can with a running system, make a complete installation to a directory. Thus it is possible, for example, to update another computer via NFS, to set up a “chroot environment”, or to perform a Linux installation on a second hard disk from the running system.

This installation mode is only intended for *special cases*. If certain devices are mounted in the directory specified, you must take care of this yourself before installing with this method.

You must especially ensure yourself that the installed system is bootable. Modifications to the `fstab` file created by YaST will also probably be necessary in the newly installed system.

Now type in the entry field which directory you want to install to.

3.4 Determining the Size of the Installation

After you have completed configuring your file system, you are ready to select the packages to be installed. From the main menu, select the option for '`Choose/Install Packages`'. The reader should be aware that during the installation process, after you have finished configuring your file system, YaST will automatically present you with the necessary menu; you will not have to select this menu option from the main menu. The menu list that appears provides a few options for managing packages, including loading, saving and creating an installation profile.

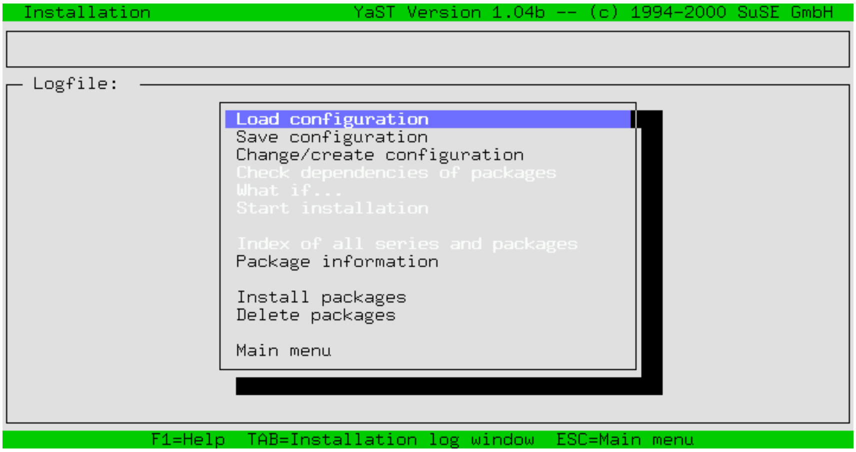

Installation
YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH
Logfile:
Load configuration
Save configuration
Change/create configuration
Check dependencies of packages
What if...
Start installation
Index of all series and packages
Package information
Install packages
Delete packages
Main menu
F1=Help TAB=Installation log window ESC=Main menu

Figure 3.16: YaST Determining the size of the installation

In addition, from this menu, you are able to start installing or to have a look at the implications of your selection of packages on the space available on your hard disk.

3.4.1 Load Configuration

There are several predefined configurations listed under '`Load Configuration`'. You can choose any of them, using the arrow keys, ↑ and ↓ to move to a predefined "configuration"; with Space you can select or unselect an item. Among them there is one labeled "Minimum system". You should choose this if you are running YaST from a floppy disk and have very little RAM. Choosing the minimal system will not prevent you from adding packages once it has been installed. YaST will perform much better after it is installed on the hard disk.

If you have created and saved your own configuration (list of packages) you can use this menu option to load your configuration. This feature is useful if you are installing an identical configuration on many systems.

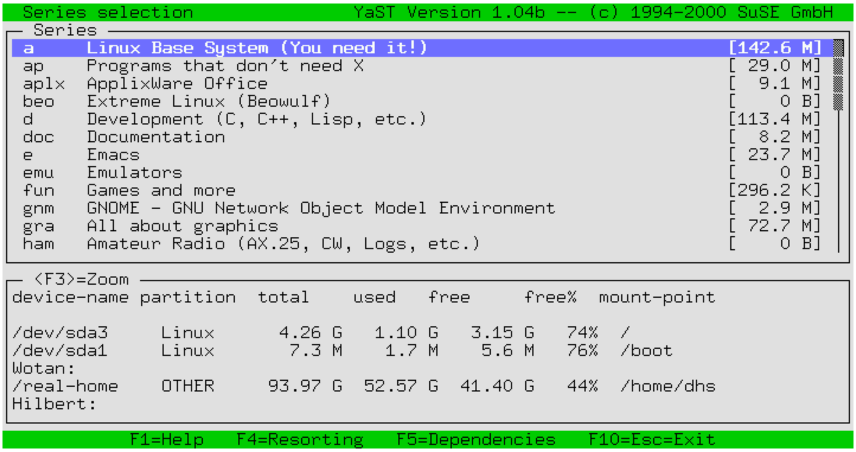
If you have already installed a system, you should be careful when you load a different configuration since all packages installed on your system which do not belong to the chosen configuration will be marked for deletion. If you want to keep the packages already installed, merely answer `no` when you are asked if you want to delete or not. If you want to delete most of those packages then you must manually unselect the ones that you do not want to delete. To unselect them, change the `'[D]'` back to a `'[i]'` by pressing Space.

3.4.2 Save Configuration

With this menu option you can save your current configuration. If YaST is running from a floppy, this will be used to save your configuration. If you have booted from a CD-ROM, you will be prompted to insert a formatted floppy. YaST will save your configuration to this.

3.4.3 Changing Your Configuration

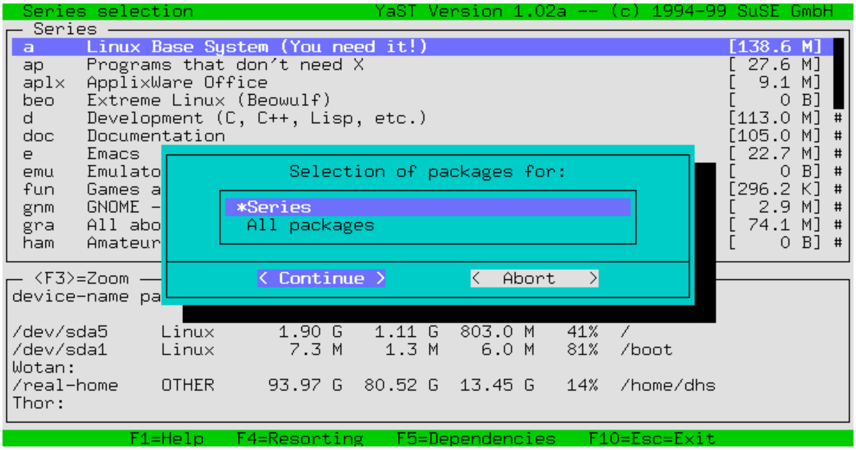
If you select '`Change/create configuration`' you will be presented with a list (Figure 3.17) of series. These series are logical groupings or categories that are used to subdivide all of the packages on the distribution. You can decide which packages to install and also select previously installed packages to be deleted. You can also change an already existing configuration (see Section 3.4.1 on the page before).



```
  Series selection                       YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH
┌ Series ──────────────────────────────────────────────────────────────────────────────┐
│ a      Linux Base System (You need it!)                                   [142.6 M] │
│ ap     Programs that don't need X                                         [ 29.0 M] │
│ aplx   ApplixWare Office                                                  [  9.1 M] │
│ beo    Extreme Linux (Beowulf)                                            [     0 B] │
│ d      Development (C, C++, Lisp, etc.)                                   [113.4 M] │
│ doc    Documentation                                                      [  8.2 M] │
│ e      Emacs                                                              [ 23.7 M] │
│ emu    Emulators                                                          [     0 B] │
│ fun    Games and more                                                     [296.2 K] │
│ gnm    GNOME - GNU Network Object Model Environment                       [  2.9 M] │
│ gra    All about graphics                                                 [ 72.7 M] │
│ ham    Amateur Radio (AX.25, CW, Logs, etc.)                              [     0 B] │
└──────────────────────────────────────────────────────────────────────────────────────┘
┌ <F3>=Zoom ───────────────────────────────────────────────────────────────────────────┐
│device-name partition  total    used    free     free%  mount-point                  │
│                                                                                      │
│/dev/sda3     Linux     4.26 G   1.10 G   3.15 G   74%   /                            │
│/dev/sda1     Linux      7.3 M    1.7 M    5.6 M   76%   /boot                        │
│Wotan:                                                                                │
│/real-home    OTHER    93.97 G  52.57 G  41.40 G   44%   /home/dhs                    │
│Hilbert:                                                                              │
└──────────────────────────────────────────────────────────────────────────────────────┘
            F1=Help   F4=Resorting   F5=Dependencies   F10=Esc=Exit
```

Figure 3.17: Selecting package series in YaST

You can move up and down the list of series with the keys ↑, ↓, Page ↑ and Page ↓ . In the lower window is a description of the current status of the different partitions on your system. The ⟵ key opens the currently highlighted series and shows you the contents of that series. The Esc key can be used to exit out of the package selection without saving any changes. If you have made changes to the package selection, the F10 key can be used to save those changes and return to the previous menu.



```
  Series selection                  YaST Version 1.02a -- (c) 1994-99 SuSE GmbH
 ┌ Series ──────────────────────────────────────────────────────────────────┐
 │ a      Linux Base System (You need it!)                      [138.6 M]   │
 │ ap     Programs that don't need X                            [ 27.6 M]   │
 │ aplx   ApplixWare Office                                     [  9.1 M]   │
 │ beo    Extreme Linux (Beowulf)                               [     0 B]  │
 │ d      Development (C, C++, Lisp, etc.)                      [113.0 M] # │
 │ doc    Documentation                                         [105.0 M] # │
 │ e      Emacs   ┌──────────────────────────────────────────┐  [ 22.7 M] # │
 │ emu    Emulato │       Selection of packages for:         │  [     0 B] # │
 │ fun    Games a │    ┌────────────────────────────────┐    │  [296.2 K] # │
 │ gnm    GNOME - │    │*Series                         │    │  [  2.9 M] # │
 │ gra    All abo │    │ All packages                   │    │  [ 74.1 M] # │
 │ ham    Amateur │    └────────────────────────────────┘    │  [     0 B] # │
 └───────────────┤                                          ├──────────────┘
 ┌ <F3>=Zoom ─── │       < Continue >        <  Abort   >   │──────────────┐
 │ device-name pa└──────────────────────────────────────────┘              │
 │                                                                          │
 │ /dev/sda5     Linux      1.90 G    1.11 G   803.0 M   41%   /            │
 │ /dev/sda1     Linux       7.3 M     1.3 M     6.0 M   81%   /boot        │
 │ Wotan:                                                                   │
 │ /real-home    OTHER     93.97 G   80.52 G   13.45 G   14%   /home/dhs    │
 │ Thor:                                                                    │
 └──────────────────────────────────────────────────────────────────────────┘
            F1=Help    F4=Resorting    F5=Dependencies    F10=Esc=Exit
```

Figure 3.18: Re-sorting packages in YaST

The F4 key can be used to select alternate views of the packages(Figure 3.18). You have the choice to view, along with the '`series`' (which is the default) and '`RPM groups`' '`All Packages`':

'`Series`' The usual categorization according to series.

'`RPM groups`' Order according to RPM group defaults.

'`All packages`' '`All packages (no sources)`'; '`Base system packages`' and '`All sources`'.

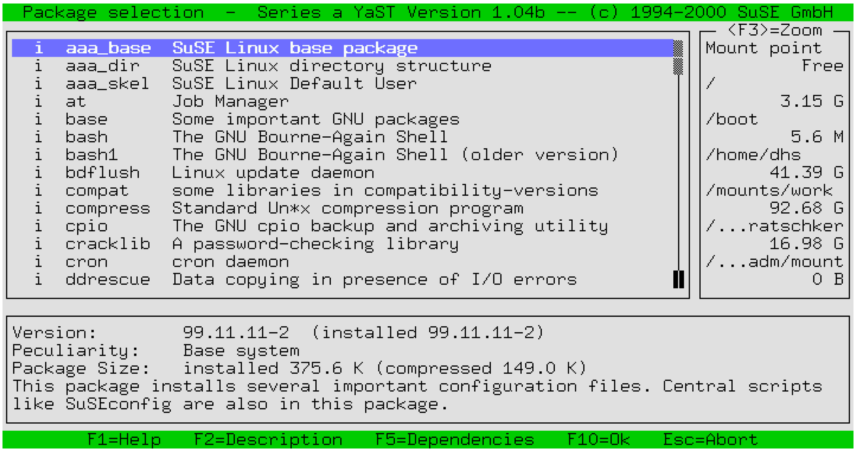
If you press ↵ on a series you will be put into 'package selection' mode for that series (Figure 3.19 on the next page shows the contents of the series `a`). If you have loaded a configuration already, the packages belonging to this configuration are marked with an `'X'`. In the bottom window you can see a short description of the highlighted package. In the right hand window you will see the amount of space that is required for your current configuration.

These values are updated as you select and deselect packages. The status windows can be enlarged with the F3 and F2 keys respectively. If you are installing your system for the first time and find that your current partition configuration does not provide enough space for the packages that you want to install, you can go back to the beginning of the installation and recreate the partitions as necessary. If you are trying to add packages to an existing system and find that you do not have enough space, you will need to find another solution (add another hard disk or perhaps delete some unwanted packages).

Each package name is preceded by its current state:

`'[ ]'` indicates that this package is not yet installed

`'[X]'` indicates that this package will be installed



```
 Package selection  -  Series a YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH
                                                                  <F3>=Zoom
  i  aaa_base   SuSE Linux base package                          Mount point
  i  aaa_dir    SuSE Linux directory structure                         Free
  i  aaa_skel   SuSE Linux Default User                          /
  i  at         Job Manager                                          3.15 G
  i  base       Some important GNU packages                      /boot
  i  bash       The GNU Bourne-Again Shell                            5.6 M
  i  bash1      The GNU Bourne-Again Shell (older version)       /home/dhs
  i  bdflush    Linux update daemon                                 41.39 G
  i  compat     some libraries in compatibility-versions         /mounts/work
  i  compress   Standard Un*x compression program                   92.68 G
  i  cpio       The GNU cpio backup and archiving utility        /...ratschker
  i  cracklib   A password-checking library                         16.98 G
  i  cron       cron daemon                                      /...adm/mount
  i  ddrescue   Data copying in presence of I/O errors                  0 B

Version:         99.11.11-2  (installed 99.11.11-2)
Peculiarity:     Base system
Package Size:    installed 375.6 K (compressed 149.0 K)
This package installs several important configuration files. Central scripts
like SuSEconfig are also in this package.

        F1=Help   F2=Description   F5=Dependencies   F10=Ok   Esc=Abort
```

Figure 3.19: Selecting packages to install: series `a1`, base system

`‘[i]’` indicates that this package is already installed

`‘[D]’` indicates that this package will be deleted

`‘[R]’` indicates that this package will be replaced

By pressing ↵ you can change the status of the package. If the package has not already been installed, pressing ↵ will change the status from `‘[ ]’` to `‘[X]’` and back again. If the package has already been installed, pressing ↵ will toggle the status between `‘[i]’`, `‘[R]’` and `‘[D]’`. F6 causes the corresponding sources package to be installed as well.For information on source packages, see also Section 11.3.4 page 289 . As stated earlier, if you want to save changes that you have made in the package selection, use the F10 key. If you do not want to save any changes, use the Esc key.

If, by any chance, you run out of space on your partitions, you can easily switch back to the main menu and repartition your hard disk. The actual partitioning will take place only after you have left YaST.

If you are running YaST from a floppy, only install a minimal system. After you have completed the minimal install, go back and install the remaining packages. With the YaST features described above you have all the tools necessary to test out various desired configurations and their hard disk requirements.

3.4.4 What if...

This menu item was created to give you a chance to review all of the changes that you have made to a configuration before committing them. Since YaST can be used to install, uninstall and replace (update) packages and there are so many packages on the distribution, this option can be a useful way to verify that you have not unintentionally marked packages for deletion or installation.

3.4.5 Start Installation

This menu option will start the installation of packages. All of the packages that you have selected for installation are copied from the installation medium, uncompressed and written to the appropriate location on your hard disk.

If you have changed the partition tables during this YaST session, installation is not yet possible. After changing the partition tables, your computer needs to be rebooted in order to write this modified partition data to the partition tables. This can only be done after leaving YaST.

During installation, status messages will be shown in the lower window. The upper window informs you about the current package being installed.

After installation, if you want, you can jump back to the installation log window by pressing the Tab key. This will allow you to view any messages that may have appeared during the installation of a particular package.

3.4.6 Checking Package Dependencies

YaST can be used to check package dependencies. Some packages are dependent on the presence of other packages. The converse is also true. Some packages should not be installed with other packages. YaST will check the packages marked for installation against the packages already installed on your system. YaST will also take into consideration packages that are marked for deletion. All unresolved package dependencies are displayed in a list.

The Boolean logic that is used to define and check these dependencies is described below.

`AND`	means that if this package is installed, all listed packages should be installed too. For example, if you install the compiler, you will also need the include and lib files.
`OR`	means that if you install this package, at least one of the listed packages should be installed as well.
`EXCL`	means that if this package is installed, none of the listed packages should be installed.

Table 3.1: Package dependencies

3.4.7 Index of all Series and Packages

This option simply provides a list of all packages included on the distribution. Those packages marked with an `'*'` are either already installed or have been selected for installation. This option gives you a quick overview of your installation selection.

3.4.8 Searching for Packages

In the event that you cannot find a particular package, but know where it is on the SuSE Linux distribution, you can use the menu option '`Package Information`' to find it. You may either search through the installation medium, or through the entire system, or both. There is even an option for doing a case sensitive search. YaST looks for the string of text that you have entered in the search field and presents you with a list of relevant packages.

3.4.9 Installing Packages

The '`Install packages`' menu option is useful for installing packages that are not on the distribution. Most importantly, this menu option can be used to update packages directly from our FTP server `ftp.suse.com`, provided that you have some form of Internet access. The packages on the FTP server are typically bug fixes or newer versions. The '`Install packages`' menu option can also be used to install packages directly from your hard disk, a disk, or another source medium. Such packages could include, but are not limited to, packages that you have received from an alternate site, packages that you have created yourself, or simply packages that you have downloaded from our FTP server and saved on your hard disk. A couple of formats are supported: tar archives (`.tgz`), RPM packages (`.rpm`, `.spm` and `.src.rpm`) as well as special patch packages (`.pat`) that are available, if required, from our FTP server.

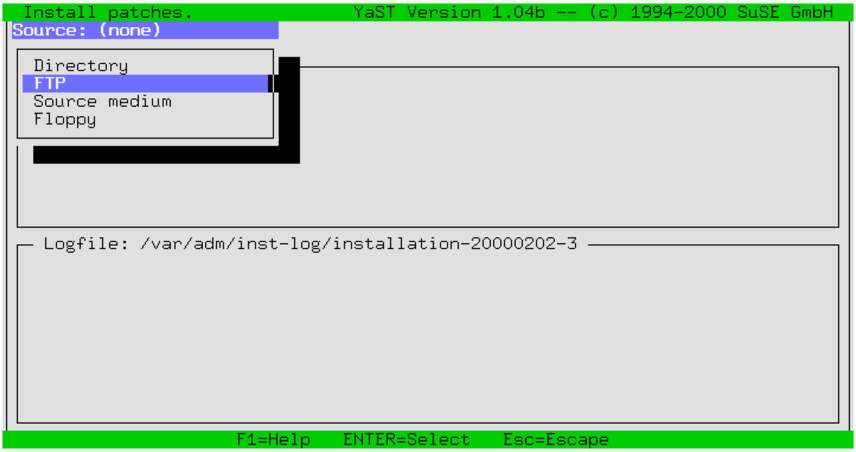
The installation procedure is divided into three steps (press F1 for complete instructions):

- Selection of the installation medium
- Selection of the packages offered
- Installation of the packages

First you select the package medium by entering '`Source:`' and pressing ⏎. The following options are available: '`Directory`', '`FTP`', '`Source medium`' and '`Floppy`'. You may need to change the default path (if you have selected '`Directory`' or '`FTP`'). YaST will connect to the source medium when you press ⏎. Thereafter, provided that you have selected the correct directory, you will be presented with a list of packages to be installed. With '`FTP`' it is thus possible to install directly from the Internet. The address is `ftp.suse.com:/pub/suse/i386/update/7.2` (refer to Figure 3.20 on the next page). It is not yet possible to install packages via a "proxy"; you need to have *direct* access to the appropriate ftp server.

Tip

If you get an error message such as `"530 User ftp access denied"`, this means that too many people are currently logged on. You will have to try again later.



```
 Install patches.                YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH
Source: (none)

  Directory
  FTP
  Source medium
  Floppy

 Logfile: /var/adm/inst-log/installation-20000202-3

                      F1=Help   ENTER=Select   Esc=Escape
```

Figure 3.20: Installing packages in YaST via FTP

First, find the directory that has the `rpm` package that you want to install. Then you can select the packages to be installed with Space. Once the package has been selected for installation, press F10 to install it. The package is first copied to the `/tmp/ftp`<`processID`> directory and then installed. If something goes wrong, you can still install it manually(see Section 11.3.2 page 286).

3.4.10 Deleting Packages

Selecting '`Deleting Packages`' will provide you with a list of the packages that are currently installed. This list includes "foreign" packages – packages which are not part of the SuSE distribution. YaST cannot update these packages or check for any dependencies unless they were installed in an RPM format.

It is easiest to replace these packages with packages from the SuSE distribution. Just select the packages that you need to replace. A short description is displayed for each package if you press F2, but only if that package contains description information. Press Space to select a package to be deleted, and F10 to delete it.

Afterwards you can reinstall those packages from the SuSE CD's.

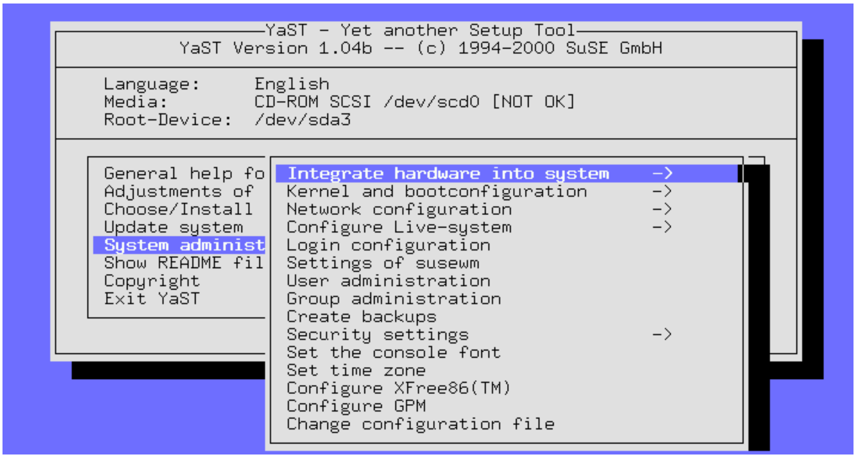
3.5 Updating the System

You should only use the menu item '`Update System`' if your base system is a fairly recent one – that is, fitting in with the source medium. If there is doubt, YaST will tell you in no uncertain terms. You can begin a system update as described in Chapter 11 page 279; individual packages can be updated with YaST as explained in Section 3.4.9 on the facing page.

3.6 System Administration

In addition to assisting you with the installation, YaST is a powerful tool for performing system administration.

Select '`System administration`' from the main menu (see Figure 3.21).



```
                      YaST - Yet another Setup Tool
               YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH

     Language:     English
     Media:        CD-ROM SCSI /dev/scd0 [NOT OK]
     Root-Device:  /dev/sda3

     General help fo  Integrate hardware into system      ->
     Adjustments of   Kernel and bootconfiguration        ->
     Choose/Install   Network configuration               ->
     Update system    Configure Live-system               ->
     System administ  Login configuration
     Show README fil  Settings of susewm
     Copyright        User administration
     Exit YaST        Group administration
                      Create backups
                      Security settings                   ->
                      Set the console font
                      Set time zone
                      Configure XFree86(TM)
                      Configure GPM
                      Change configuration file
```

Figure 3.21: Administering your system

From the main menu in YaST (see Figure 3.1 page 52) you can choose the menu option '`System Administration`' to access the system administration features of YaST (Figure 3.21).

3.6.1 Integrating Hardware into the System

With this option you can specify the hardware that you are using. In most cases, YaST will create a link from the standard device to your particular hardware device. This will make it easier for you to access this device without having to remember the exact name of the device.

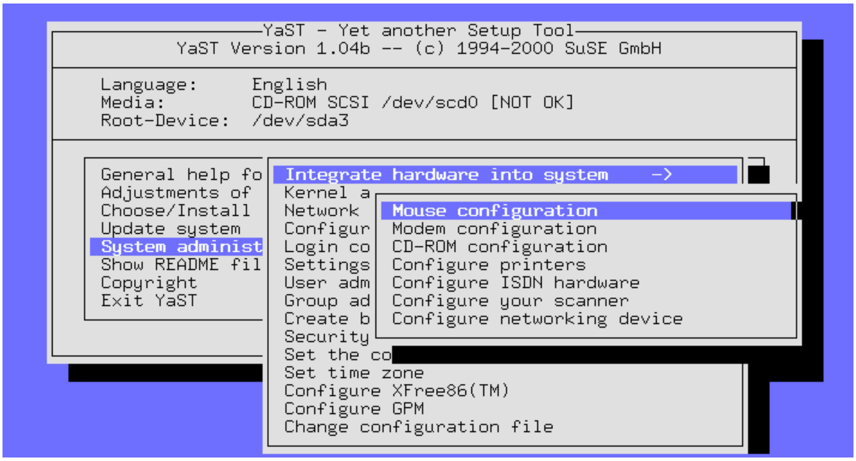
The printer configuration is a bit more complicated. This is described in the following section.

Printer configuration

Accessing a printer under Linux is not trivial. The technical background is described in detail in Chapter 8 page 205.

YaST provides an interface for easily setting up apsfilter for your printer (Figure 3.23 page 80):

Under the menu option '`Printer name`' you will be presented with a list of printers that are supported by Ghostscript. You can select your printer from this list. If your printer does not appear on this list, you can select '`Other printer`'



```
                     YaST - Yet another Setup Tool
            YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH

    Language:      English
    Media:         CD-ROM SCSI /dev/scd0 [NOT OK]
    Root-Device:   /dev/sda3

    General help fo  Integrate hardware into system    ->
    Adjustments of   Kernel a
    Choose/Install   Network   Mouse configuration
    Update system    Configur  Modem configuration
    System administ  Login co  CD-ROM configuration
    Show README fil  Settings  Configure printers
    Copyright        User adm  Configure ISDN hardware
    Exit YaST        Group ad  Configure your scanner
                     Create b  Configure networking device
                     Security
                     Set the co
                     Set time zone
                     Configure XFree86(TM)
                     Configure GPM
                     Change configuration file
```

Figure 3.22: Hardware configuration

and then the option '<userdefined>'. You will be presented with a screen where you can enter your printer type. The command **gs -h** will display a list of printers that are supported by the currently installed version of Ghostscript. If your printer appears on this list, you can enter it as the '<user-defined>' printer type. The configuration of uniprint drivers can be best accomplished with YaST2 or the SETUP (**lprsetup**) program. This program is part of the apsfilter package. Please see Section 8.6 page 217 for more information.

Most normal printers are connected to the parallel port (printer port) or USB port. In most cases /dev/lp0 will be the correct port – that is, the first parallel port.

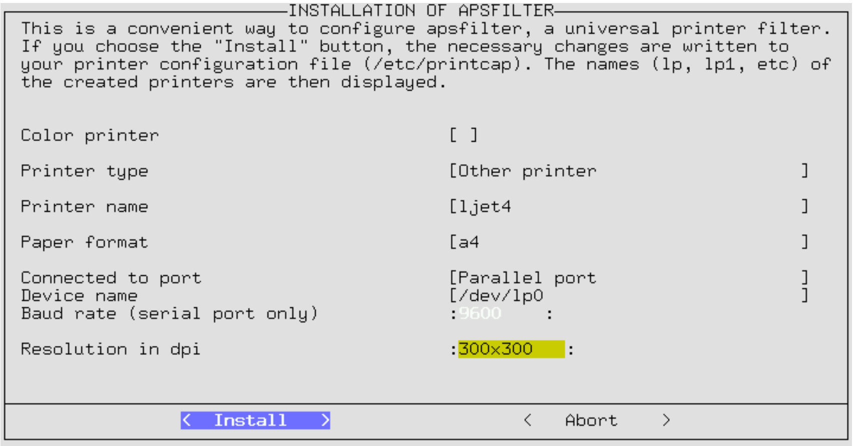
Tip

If in doubt, first set the following values in the BIOS for the parallel port:

- IO address 378 (hexadecimal)
- interrupt 7
- as mode: normal (or SPP)
- and switch off DMA (should be switched off in normal mode)

These are the normal standard values.

The settings for the menu option 'Resolution in dpi' should be verified. If you are unsure, you will have to refer to your printer documentation.

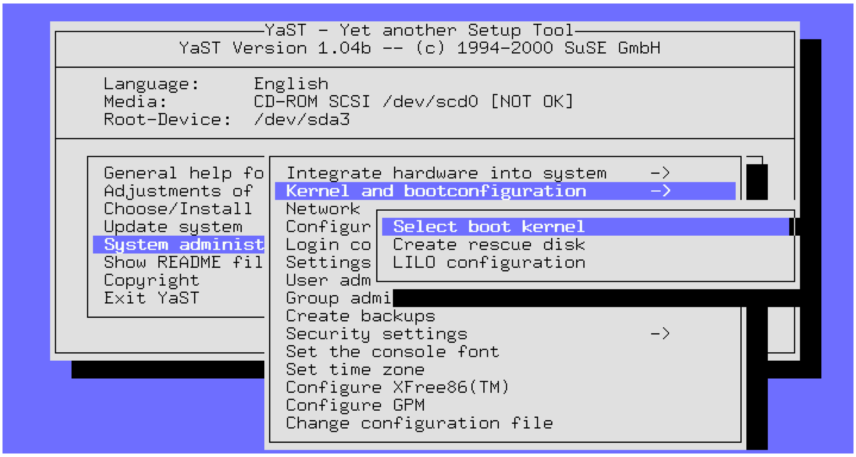


```
INSTALLATION OF APSFILTER
This is a convenient way to configure apsfilter, a universal printer filter.
If you choose the "Install" button, the necessary changes are written to
your printer configuration file (/etc/printcap). The names (lp, lp1, etc) of
the created printers are then displayed.

Color printer                           [ ]

Printer type                            [Other printer                    ]

Printer name                            [ljet4                            ]

Paper format                            [a4                               ]

Connected to port                       [Parallel port                    ]
Device name                             [/dev/lp0                         ]
Baud rate (serial port only)            :9600      :

Resolution in dpi                       :300x300   :

                < Install >                       <    Abort    >
```

Figure 3.23: Setting up apsfilter with YaST

3.6.2 Kernel and Boot Configuration

This set of sub-menus can be used to configure your system’s boot setup as well as which kernel you want to use:



```
YaST - Yet another Setup Tool
YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH

Language:     English
Media:        CD-ROM SCSI /dev/scd0 [NOT OK]
Root-Device:  /dev/sda3

General help fo | Integrate hardware into system     ->
Adjustments of  | Kernel and bootconfiguration       ->
Choose/Install  | Network
Update system   | Configur | Select boot kernel
System administ | Login co | Create rescue disk
Show README fil | Settings | LILO configuration
Copyright       | User adm
Exit YaST       | Group admi
                | Create backups
                | Security settings                  ->
                | Set the console font
                | Set time zone
                | Configure XFree86(TM)
                | Configure GPM
                | Change configuration file
```

Figure 3.24: Kernel and boot configuration

Selecting a boot kernel

You can install one of the available precompiled kernels (from the SuSE-CD) with the menu option ‘`Select boot kernel`’, in the event that you want a different kernel than the one you chose during installation. In addition to installing the kernel that you select, YaST will also copy the kernel configuration file (`.config`) to the kernel source directory (`/usr/src/linux`).

Select the appropriate kernel for your computer. Then, as a rule, you should have LILO installed again, using YaST, if you have already booted successfully with LILO. Otherwise see below, the guide on configuring LILO.

Advanced Linux users might want to consider recompiling their own kernel (see Chapter 9 page 237). One advantage of recompiling the kernel is that it can be customized to fit your exact hardware configuration, and it can be made smaller and faster. If you are new to Linux it is *highly* recommended that you use the precompiled standard kernels. Only if you are using the standard kernel you are qualified for “Installation support”. In other words, by recompiling your own kernel you no longer qualify for “Installation support”.

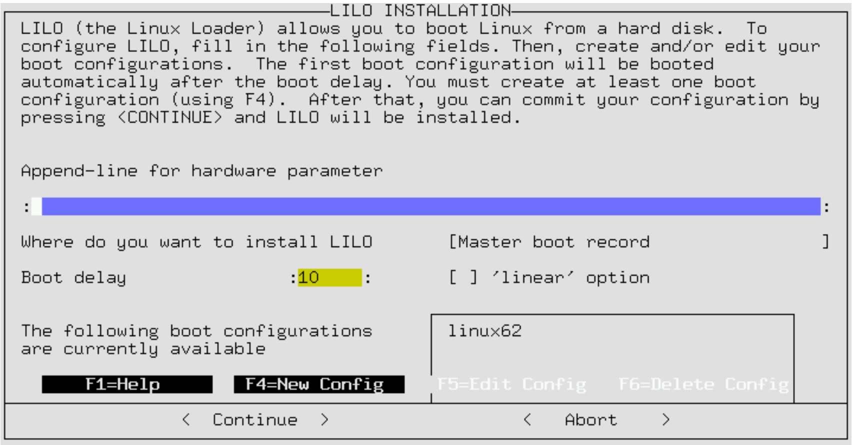
Creating a rescue disk

An “emergency” or “Rescue disk” is also a good idea, in the event that your computer will not start at all – see Section 12.6 page 302 for details on how to use this floppy disk).

Configuring ‘LILO’

YaST also provides a front-end to LILO (Linux Loader) to ensure correct configuration (Figure 3.25). LILO can be used to boot other operating systems such as OS/2, DOS or Windows 95/98, but with Windows NT you should be careful.

Additional information on the individual configuration options is available in Chapter 4 page 93.



```
LILO INSTALLATION
LILO (the Linux Loader) allows you to boot Linux from a hard disk.  To
configure LILO, fill in the following fields. Then, create and/or edit your
boot configurations.  The first boot configuration will be booted
automatically after the boot delay. You must create at least one boot
configuration (using F4).  After that, you can commit your configuration by
pressing <CONTINUE> and LILO will be installed.


Append-line for hardware parameter

:                                                                          :

Where do you want to install LILO       [Master boot record               ]

Boot delay                    :10    :  [ ] 'linear' option


The following boot configurations        linux62
are currently available
```

Figure 3.25: LILO: Installation

The ‘append line for kernel parameters’ usually remains empty unless you already need to use additional parameters for booting; only if this is the case should you enter these parameters here (but *without* the preceding kernel name linux!).

Details can be found in Section 4.4.2 page 101.

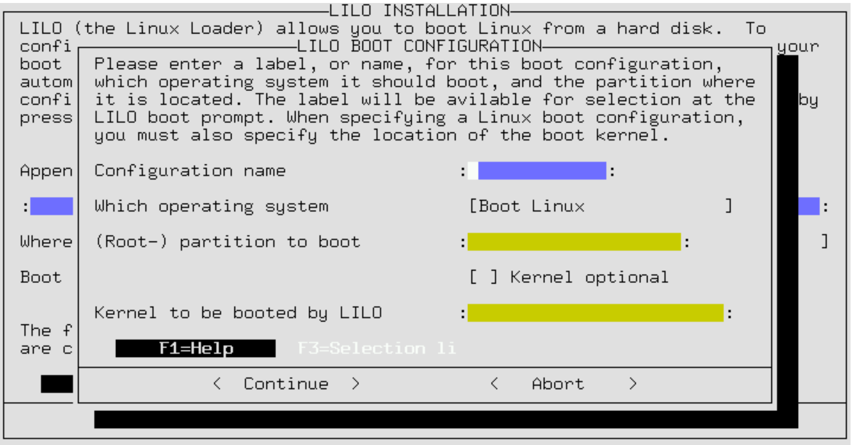
‘Where do you want to install LILO?’: If Linux is the *only* operating system on your computer, then Master boot record is the correct choice. If you are using “another” boot manager, select
Boot sector of the /boot partition. If you have not created a /boot partition, select
boot-sector of the root partition. The choice On floppy disk speaks for itself. The technical background of these choices is given in Section 4.3 page 97.

‘Boot delay’: This entry is in seconds. 10 seconds is a reasonable value to use.

‘"linear" Option’: In most cases this option is *not* necessary.

For additional information please see Section 4.4.2 page 101.

The ‘F4=New Config’ menu option can be used to create a “new configuration”. It is recommended that you use the standard configuration linux as the configuration name. To edit an existing configuration, use the ‘F5=Edit config’ menu option. When using either of these keys, YaST will present you with a configuration screen, as shown in Figure 3.26. The individual options on this screen are described below:



```
LILO INSTALLATION
LILO (the Linux Loader) allows you to boot Linux from a hard disk.  To
confi   LILO BOOT CONFIGURATION                                     your
boot    Please enter a label, or name, for this boot configuration,
autom   which operating system it should boot, and the partition where
confi   it is located. The label will be avilable for selection at the   by
press   LILO boot prompt. When specifying a Linux boot configuration,
        you must also specify the location of the boot kernel.

Appen   Configuration name                  :            :

:       Which operating system              [Boot Linux        ]          :

Where   (Root-) partition to boot           :               :             ]

Boot                                        [ ] Kernel optional

        Kernel to be booted by LILO         :                    :
The f
are c     F1=Help        F3=Selection li

              <  Continue  >           <  Abort  >
```

Figure 3.26: LILO: Boot configuration

‘Configuration name’: You can choose as you wish, but linux is a good name for your first configuration.

‘Which operating system’: Your three choices are Boot linux, Boot DOS/Win - this option is for Windows 95/98 as well – and Boot OS/2. ‘(root-) par-

`tition to boot`': This detail is pre-defined. As a rule, you should not change anything here. Press '`F3`' and choose the appropriate partition.

'`Kernel optional`': Only select this option if the kernel is *not permanently* available. This would be chosen if you wanted to boot to a test kernel only once.

'`Kernel to be booted by Lilo`': here as well you do not need to change anything. The usual location in SuSE Linux is `/boot/vmlinuz`; By pressing '`F3`' you can browse through the directories and select the kernel. With '`Continue`' LILO is installed, with '`Cancel`' you can stop the installation of LILO.

Tip

If you are planning on compiling your own kernel, we recommend that you create a second configuration in LILO. The purpose of this new configuration is a backup option in case your newly compiled kernel does not boot. Create a new configuration and call it `old`, and configure it to use the `/boot/vmlinuz.old` kernel. You should also copy the current kernel `/boot/vmlinuz` to `/boot/vmlinuz.old`. When configuring this option, you should also enable the '`Kernel optional`' option. Now, in the event that you have compiled a new kernel (see Chapter 9 page 237) which for some odd reason will not boot, you still have a way to boot to your original kernel.

3.6.3 Network Configuration

General network configurations can be done with YaST as well(see Figure 3.27 on the following page). Even if you are not connected to a network, it is a good idea to configure your network. Many programs depend on a correctly configured network to be able to function properly.

'`Network base configuration`' This is where you can configure the IP-Address(es) for the appropriate network interface (network card, PPP, or ISDN, etc.) (Figure 3.28 page 85).

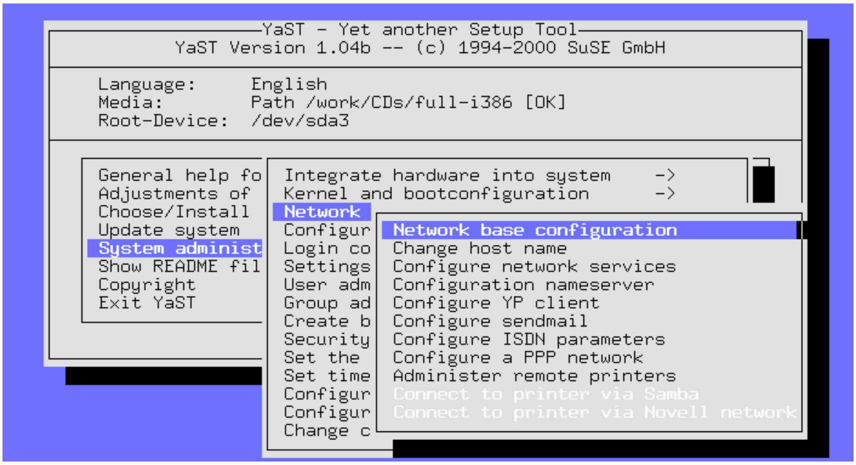
Here you can also select the services DHCP or BOOTP via '`F3=Auto-IP`', to use an automatic IP configuration.

Using '`F6=IP-Address`' gives you the ability to change the '`MTU`'. In case of doubt keep the entry blank or use the provided default value.

MTU is the "Maximum Transfer Unit" and sets the maximum size of packages used on that network device.

Sometimes your provider forces you to use a special value for '`MTU`'. The standard value (and maximum) is '`1500`'. In case the connection to the provider is pretty bad, you should reduce the value of '`MTU`'.

'`Change hostname`' The hostname and domain name of your computer can be set here.



```
                    YaST - Yet another Setup Tool
            YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH

   Language:      English
   Media:         Path /work/CDs/full-i386 [OK]
   Root-Device:   /dev/sda3

   General help fo  Integrate hardware into system     ->
   Adjustments of   Kernel and bootconfiguration       ->
   Choose/Install   Network
   Update system    Configur  Network base configuration
   System administ  Login co  Change host name
   Show README fil  Settings  Configure network services
   Copyright        User adm  Configuration nameserver
   Exit YaST        Group ad  Configure YP client
                    Create b  Configure sendmail
                    Security  Configure ISDN parameters
                    Set the   Configure a PPP network
                    Set time  Administer remote printers
                    Configur  Connect to printer via Samba
                    Configur  Connect to printer via Novell network
                    Change c
```

Figure 3.27: Network configuration

‘`Configure network services`’ With this option you can configure basic network services such as: inetd (for telnet, ftp, printing etc..) portmap (for NFS server and NIS) and rpc.nfsd (for NFS server).

‘`Configuration Nameserver`’ It is possible to specify one or more nameservers. (see Section 19 page 321 for more information).

‘`Configure YP client`’ This option will only be active if you have installed the package `ypclient`, series `n`.

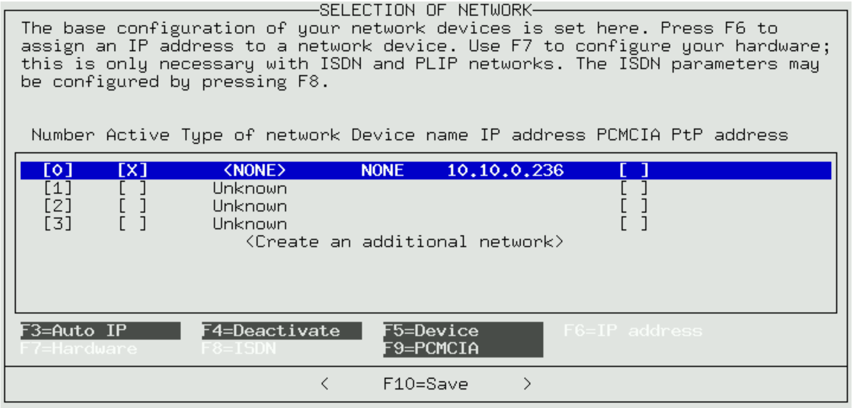
‘`DHCP Client`’ This option will only be active if the package `dhclient`, series `n` has been installed.

‘`Configure Sendmail`’ With this option it is possible to install a configuration file for sendmail. There are a few configuration files available that will suit most needs.

‘`Administer remote printers`’ With this option you can configure your machine to be able to access a printer which is on a TCP/IP network. The configuration options in this screen are relatively self-explanatory. ‘`Name of printer`’ refers to the name by which the local machine refers to the printer (a good choice would be `remote`). ‘`Spool directory`’ will be automatically configured according to the name given above. ‘`Server name`’ is the IP address or the name of the print server. ‘`Name of printer`’ is the name by which the print server refers to the printer (in most cases this will be `lp`).

If you need a prefilter for this printer, please refer to Section 8.8 page 224.

‘`Connect to printer via Samba`’ This option will only be active if the package `samba`, series `n`, has been installed. With this menu item it is possible to connect to a printer that is connected to a Windows computer.



```
────────────────────SELECTION OF NETWORK────────────────────
 The base configuration of your network devices is set here. Press F6 to
 assign an IP address to a network device. Use F7 to configure your hardware;
 this is only necessary with ISDN and PLIP networks. The ISDN parameters may
 be configured by pressing F8.


  Number Active Type of network Device name IP address PCMCIA PtP address

    [0]     [X]       <NONE>          NONE      10.10.0.236     [ ]
    [1]     [ ]      Unknown                                    [ ]
    [2]     [ ]      Unknown                                    [ ]
    [3]     [ ]      Unknown                                    [ ]
                        <Create an additional network>


 F3=Auto IP         F4=Deactivate      F5=Device          F6=IP address
 F7=Hardware        F8=ISDN            F9=PCMCIA

                          <       F10=Save        >
```

Figure 3.28: Network base configuration

'`Connect to a printer via Novell network`' This option will be active only if the package `ncpfs`, series `n` has been installed.

'`Configure ISDN parameters`' This option will only be active if the package `i4l`, series `n`, has been installed. Follow the detailed menus that are available.

'`Configure a PPP network`' Here PPP can be set up.

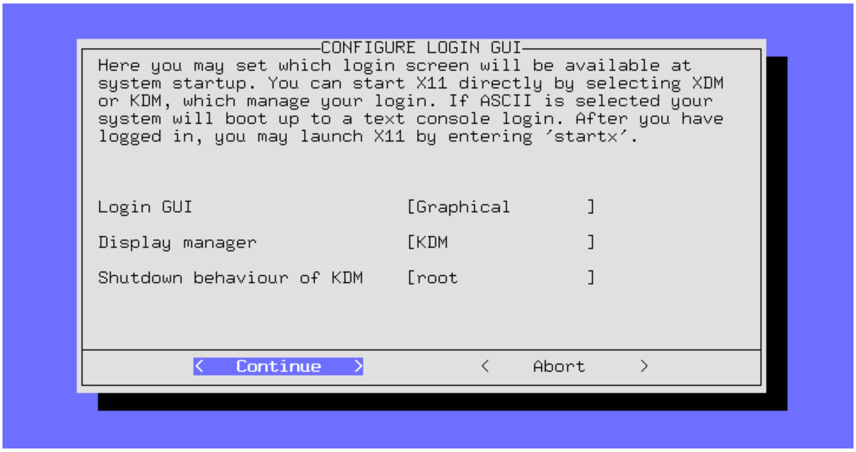
3.6.4 Login configuration

Here you can set whether the system will boot to a text mode or to a graphical mode with the X Window System. If you want to use the graphical login, you have two choices: XDM or KDM. If you choose KDM you can also select which users are permitted to **`shutdown`** the system. The other login option is via the text console. With this option you can start the X Window System with the command startx (Figure 3.29 on the following page).

Note

If you are not absolutely sure that the X Windows System is configured correctly, you should not activate either of the graphical logins. Please test whether you can start the X Windows System first, from the console, before activating the display manager.

If you do configure your system to start with a graphical login, the file `/etc/inittab` will be configured to set Runlevel 5 as the default runlevel (see Section 19 page 325). If Runlevel 5 serves another purpose on your machine, then *neither* XDM nor KDM will be activated.

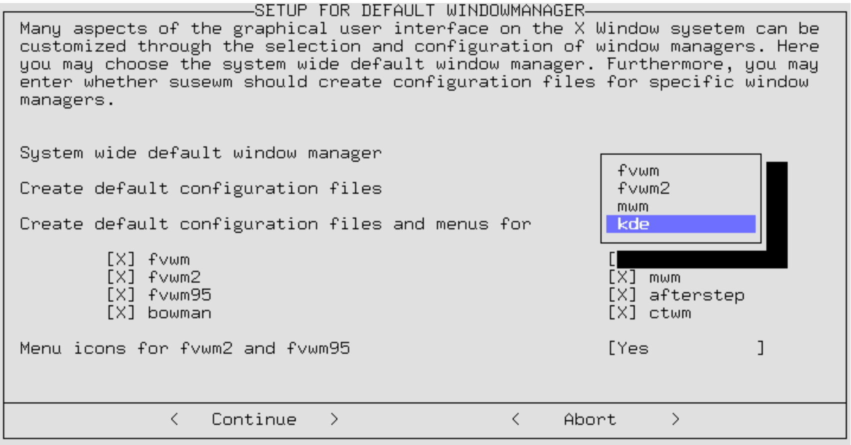


```
CONFIGURE LOGIN GUI
Here you may set which login screen will be available at
system startup. You can start X11 directly by selecting XDM
or KDM, which manage your login. If ASCII is selected your
system will boot up to a text console login. After you have
logged in, you may launch X11 by entering 'startx'.


Login GUI                     [Graphical         ]

Display manager               [KDM               ]

Shutdown behaviour of KDM     [root              ]
```

Figure 3.29: Login Configuration

3.6.5 Settings for susewm (the Window Manager)

With this menu option you can set the "Graphical Desktop".



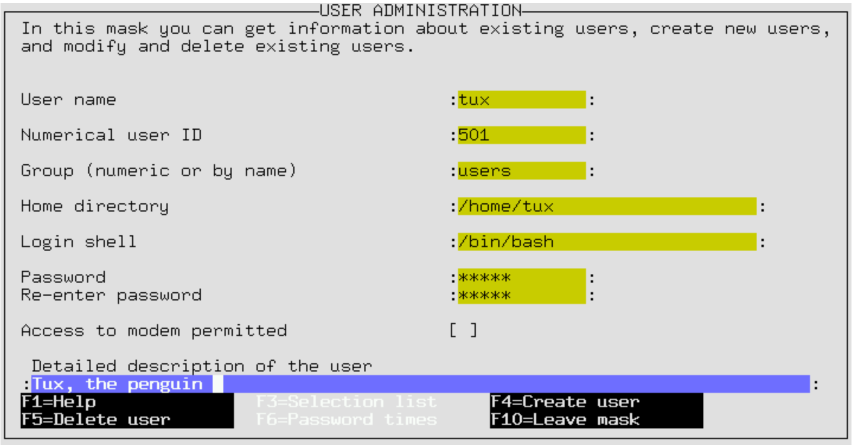
```
SETUP FOR DEFAULT WINDOWMANAGER
Many aspects of the graphical user interface on the X Window sysetem can be
customized through the selection and configuration of window managers. Here
you may choose the system wide default window manager. Furthermore, you may
enter whether susewm should create configuration files for specific window
managers.


System wide default window manager                        fvwm
                                                          fvwm2
Create default configuration files                        mwm
                                                          kde
Create default configuration files and menus for

        [X] fvwm                                          [
        [X] fvwm2                                         [X] mwm
        [X] fvwm95                                        [X] afterstep
        [X] bowman                                        [X] ctwm

Menu icons for fvwm2 and fvwm95                           [Yes          ]
```

Figure 3.30: Setting up susewm

Please select your default window manager. You could also tell susewm which configuration files it should create and maintain.

3.6.6 User Administration

YaST provides a convenient means of creating, deleting and maintaining user accounts. The menu option for managing users is titled '`User Administration`' and is a sub-menu of '`System Administration`'.



```
                           USER ADMINISTRATION
 In this mask you can get information about existing users, create new users,
 and modify and delete existing users.

 User name                                 :tux        :

 Numerical user ID                         :501        :

 Group (numeric or by name)                :users      :

 Home directory                            :/home/tux                :

 Login shell                               :/bin/bash                :

 Password                                  :*****      :
 Re-enter password                         :*****      :

 Access to modem permitted                 [ ]

  Detailed description of the user
 :Tux, the penguin                                                          :
 F1=Help              F3=Selection list     F4=Create user
 F5=Delete user       F6=Password times     F10=Leave mask
```

Figure 3.31: User Administration with YaST

Here you can enable '`Access to Modem`'; if you do this you are added to the groups '`uucp`' and '`dialout`'. All members of these groups are able to connect and disconnect by means of PPP connections.

When new user accounts are created, default configuration files are copied from the `/etc/skel` directory into the home directory of the new user. This is a process by which the personal configuration files can automatically be created. Users are free to change those configuration files in their home directory as they wish, of course.

When creating or deleting user accounts, YaST runs the following two scripts:

- When creating a user account, the script `/usr/sbin/useradd.local` is run, if it exists. This script will create the necessary entries in the `/etc/passwd` and `/etc/shadow` files. The home directory is created and the files in `/etc/skel` are copied to it.
- Before deleting a user account, the script `/usr/sbin/userdel.local` is called. The relevant lines in `passwd` and `shadow` are removed. The home directories will remain.

Both scripts can take the username as a parameter. If additional information is needed (user ID, login shell, home directory), it can be found in the `/etc/passwd` file.

If you are an advanced user and you do not want to use YaST to create user accounts, you have the programs useradd and userdel at your disposal.

Notes on PAM

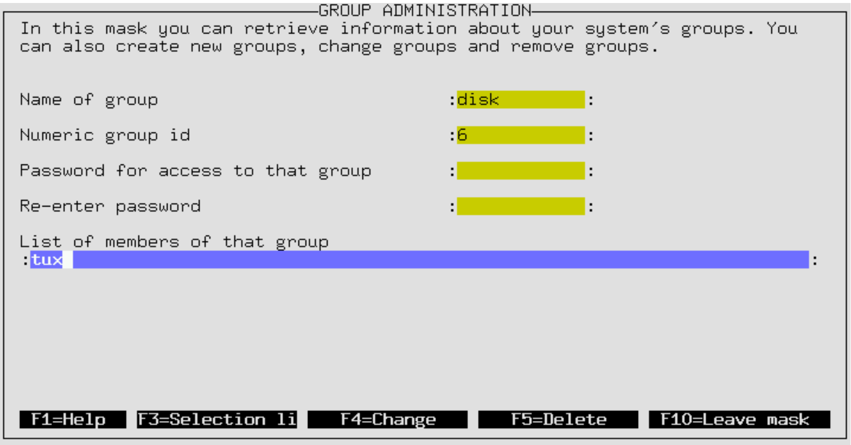
The configuration files on PAM *Pluggable Authentication Modules* can be found in `/etc/pam.d`. Documentation for programmers and system administrators is located in /usr/share/doc/packages/pam.

SuSE Linux is able to handle MD5 passwords. With MD5 encryption, passwords can be longer than 8 characters (up to 128 characters). Since MD5 encryption is *not* compatible with the standard Unix **crypt()** function, most commercial Unix systems and some programs don't work with MD5 passwords. So be careful if you enable this feature.

Notes on configuration can be found in /usr/share/doc/packages/pam/md5.config.

3.6.7 Group Administration

YaST can also be used to create user groups.



```
GROUP ADMINISTRATION
In this mask you can retrieve information about your system's groups. You
can also create new groups, change groups and remove groups.

Name of group                              :disk        :
Numeric group id                           :6           :
Password for access to that group          :            :
Re-enter password                          :            :
List of members of that group
:tux                                                              :
```

Figure 3.32: Group administration with YaST

Under Linux (as with UNIX in general) each user must be assigned to at least one group. This is necessary because permissions (for certain files) depend on which group the user belongs. User groups can be used to restrict (or allow) users access to certain directories. Group passwords can also be used to restrict (or allow) access to certain directories.

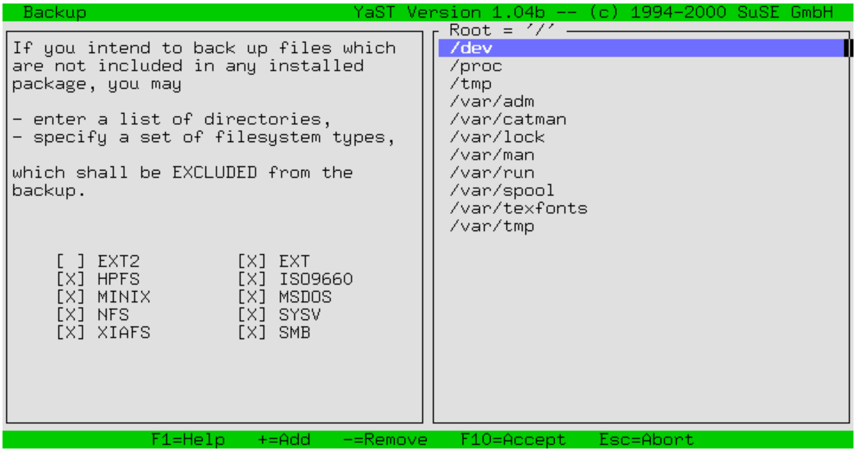
Some groups already exist under Linux, for example the user groups 'users', 'root' and many more.

The group name, such as with 'users', is just a textual representation of the group. Linux recognizes the groups according to the "GID" (Group ID). The configuration file for groups is in /etc/group.

This is just background information that most regular users do not need to know. With YaST it is easy to create groups. In the menu 'System Administration' there is a submenu for 'Group Administration'. This screen is described in Figure 3.32.

3.6.8 Creating Backups

This option helps you to back up all modified and new files and packages to a file or tape. These are configuration files in most cases.



```
 Backup                                YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH
                                                  ┌ Root = '/' ─────────────────────────┐
 If you intend to back up files which             │/dev                                 │
 are not included in any installed                │/proc                                │
 package, you may                                 │/tmp                                 │
                                                  │/var/adm                             │
 - enter a list of directories,                   │/var/catman                          │
 - specify a set of filesystem types,             │/var/lock                            │
                                                  │/var/man                             │
 which shall be EXCLUDED from the                 │/var/run                             │
 backup.                                          │/var/spool                           │
                                                  │/var/texfonts                        │
                                                  │/var/tmp                             │

     [ ] EXT2           [X] EXT
     [X] HPFS           [X] ISO9660
     [X] MINIX          [X] MSDOS
     [X] NFS            [X] SYSV
     [X] XIAFS          [X] SMB

              F1=Help   +=Add   -=Remove   F10=Accept   Esc=Abort
```

Figure 3.33: Backup with YaST—choosing directories to exclude

The dialog consists of three parts (see Figure 3.33):

1. Choosing the files to be backed up:

   Here you can tell YaST which directories should be excluded from the backup. Predefined are `/tmp` `/dev` and `/proc`. You should add mounted CD-ROM's or NFS-mounted file systems to this list. The less you want to be backed up, the faster it will run, since unnecessary comparisons with package lists are omitted. Using + and -, you can add new directories or remove them. Pressing F10 leads to the next step.

2. Searching:

   In this step, YaST searches for files which should be backed up. The number and size of the packages found are updated while searching. After this has been done, there will be a list with all the files that have been found. Here you can still de-select files by using the Space bar.

3. Entering commands:

   Here you decide how those files are going to be saved. You can give archive names, options and more.

This back up mechanism can only work if the dates of the files have not been otherwise changed. Furthermore, this function requires considerable RAM. Filenames of an ordinary CD take up to 6 MB RAM. Also, you need enough free disk space to save the backup archive. Compressing the archive will lead to a file reduced in size—approximately half of the original. The best way to do backups is to use a tape.

3.6.9 System Security Settings

Only change the defaults if you have detailed knowledge of the measures in question.

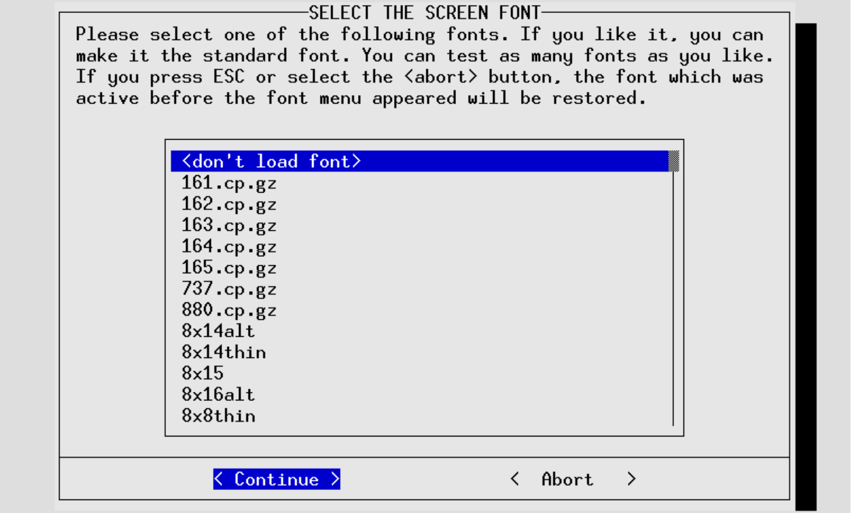
Note

If you do not want to allow any `'root'` logins via ssh, then `/etc/ssh_config` must be adjusted. In `/etc/sshd_config`, the default is set to **`PasswordAuthentication yes`**. If ssh is correctly configured, `'root'` logins are secure across a network.

3.6.10 Setting the Console Font

Here, we will focus on the screen for the font settings in the Linux console. Linux enables you to load various fonts for the console. In this settings screen, you can try out the different available fonts and apply one as your standard font. If you select '`don't load font`', the font saved on your graphics card will still be applied.

All of this, of course, does not have any effect on the fonts in a graphical interface such as KDE or Gnome. You can see the results in YaST if you have started YaST directly from the ASCII console, but not if you have started it from a kterm or an xterm.



SELECT THE SCREEN FONT

Please select one of the following fonts. If you like it, you can make it the standard font. You can test as many fonts as you like. If you press ESC or select the <abort> button, the font which was active before the font menu appeared will be restored.

```
<don't load font>
161.cp.gz
162.cp.gz
163.cp.gz
164.cp.gz
165.cp.gz
737.cp.gz
880.cp.gz
8x14alt
8x14thin
8x15
8x16alt
8x8thin
```

Figure 3.34: Setting the console font

3.6.11 Setting the time zone

The time zone has been preset during the SuSE Linux installation, but you can change or correct these settings later. Simply move the blue bar using the cursor keys ↑ and ↓ to the desired selection and then confirm with ‘Next’.

Afterwards, a window will appear which will ask you if you want to set your BIOS clock to the local time or leave it set to GMT (Greenwich Mean Time).

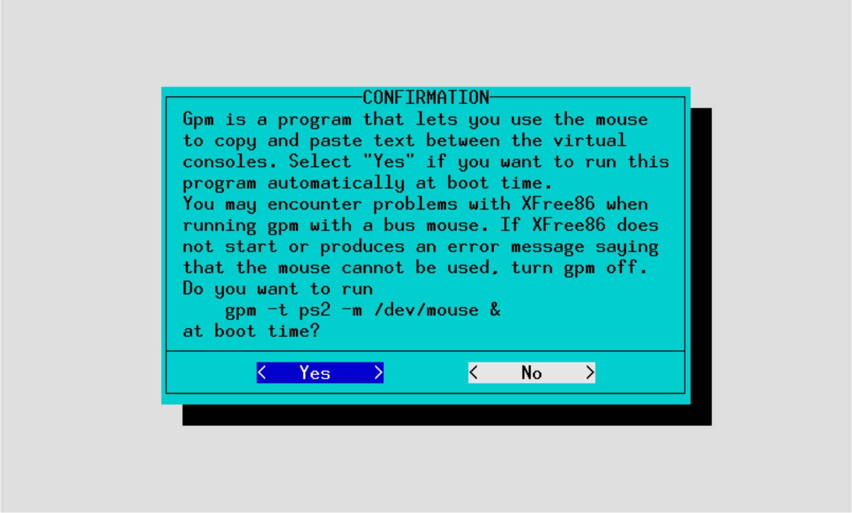
3.6.12 Configuring XFree86(TM)

The X Window System (XFree86) can be configured with different configuration tools.You should first try to use SaX. SaX is described in more detail in Section 5.3 page 128.

The technical details can be read in Chapter 5 page 125.

3.6.13 Configuring gpm

This screen controls whether to load the mouse support and with which parameters to start it. This allows you to cut and paste text between several virtual ASCII consoles. Again, this entire process is only relevant for ASCII consoles, since the mouse is controlled on the graphical interface by X11 and thus not affected by the gpm settings.



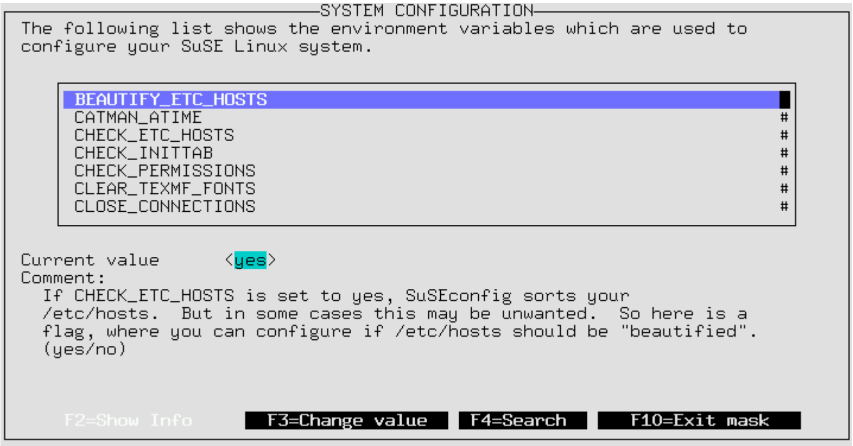
```
CONFIRMATION
Gpm is a program that lets you use the mouse
to copy and paste text between the virtual
consoles. Select "Yes" if you want to run this
program automatically at boot time.
You may encounter problems with XFree86 when
running gpm with a bus mouse. If XFree86 does
not start or produces an error message saying
that the mouse cannot be used, turn gpm off.
Do you want to run
    gpm -t ps2 -m /dev/mouse &
at boot time?
```

Figure 3.35: Configuring gpm

3.6.14 Changing the Configuration File

SuSE Linux is maintained by one central configuration file, `/etc/rc.config`, and the out-sourced files in the `/etc/rc.config.d` directory. These files are read at boot time by the boot scripts that configure your system.

It is possible to change individual variables in these files with YaST. By doing this you can make configuration changes to your system without having to know every detail about all the special configuration files that have been effected by these changes.



```
The following list shows the environment variables which are used to
configure your SuSE Linux system.

     BEAUTIFY_ETC_HOSTS
     CATMAN_ATIME
     CHECK_ETC_HOSTS
     CHECK_INITTAB
     CHECK_PERMISSIONS
     CLEAR_TEXMF_FONTS
     CLOSE_CONNECTIONS

Current value       <yes>
Comment:
  If CHECK_ETC_HOSTS is set to yes, SuSEconfig sorts your
  /etc/hosts.  But in some cases this may be unwanted.  So here is a
  flag, where you can configure if /etc/hosts should be "beautified".
  (yes/no)
```

Figure 3.36: Editing the configuration file with YaST

The menu option for changing the configuration file is under the '`System Administration`' in the menu titled '`Change Configuration file`' With the cursor, select the variable that you want to change and press the ↵ or F3 key. After you have used YaST to change one of the values the script SuSEconfig will run automatically.

If you decide to manually change this value (by editing the file `/etc/rc.config`) please remember that you must manually run the script SuSEconfig. This script will do the necessary updates to the relevant configuration files, according to the values that have been set (or changed) in `/etc/rc.config`. Additional details on this subject can be found in Section 13.6 page 317.

4 Booting and Boot Managers: LILO, loadlin, etc.

This chapter describes various methods of Booting a Linux system. To have a better understanding of what is involved, we will first illustrate some technical details of booting a PC.

4.1 Booting a PC

After turning on your computer, the first thing that happens is that the BIOS (Basic Input Output System) takes control, initializes the screen and keyboard, and tests the main memory. Until this task is completed, no external devices or external storage media are known to the system.

Once the basic system has finished its internal setup, it starts to verify the hardware around it. Date, time, and information about some of the most important external devices are read from the CMOS settings (usually referred to as the CMOS setup). After reading the CMOS, the BIOS should recognize the first hard disk (including details such as its geometry). It can then start to load the operating system (OS) from there.

To load the OS, the system loads a 512-byte data segment from the first hard disk into main memory and executes the code stored at the beginning of this segment. The instructions contained there determine the rest of the boot process. This is the reason why the first 512 bytes of the hard disk are often called the *Master Boot Record* (MBR) (MBR).

Even though the whole process is rather complicated and this description is oversimplified, it should be clear that up to this point (loading of the MBR), the boot sequence is independent of the installed operating system, is identical on all PC's, and all the PC has to access peripheral hardware is those routines (drivers) stored in the BIOS.

Master Boot Record

The layout of the MBR has been certified by an independent convention. The first 446 bytes are reserved for program code. The next 64 bytes offer space for a partition table for up to four partitions. The last two bytes have to contain a special “magic number” (AA55). Any MBR which replaces this number by a different number is rejected.

Boot Sectors

Boot sectors are the first sectors on a hard disk partition except for the extended partition which serves as a “container” for other partitions They offer 512 bytes of space and are designed to contain code which is able to launch an operating system on this partition. Boot sectors of formatted DOS, Windows, and OS/2 partitions do exactly that. In contrast, Linux boot partitions are empty at the very start. A Linux partition cannot be started directly, although it may contain a kernel and a valid root file system.

A valid boot sector follows the conventions and enters the “magic number” of the MBR into the last two bytes.

Booting DOS or Windows 95/98

The MBR contains information that determines which partition of a hard disk is “active”, i. e., which partition should be searched for the operating system to be booted. Therefore, DOS has to be installed on the first hard disk. The executable code in the MBR (“first stage boot loader”) tests whether the marked partition contains a valid boot sector.

If this is the case, the “second stage boot loader” can be started from there. DOS system programs can now be loaded and you will see the usual DOS prompt.

In DOS, only primary partitions can be marked active. Therefore, you cannot use logical partitions inside an extended partition as bootable DOS partitions.

4.2 Boot Concepts

The simplest boot concept affects only one machine with one operating system installed. A widely deployed PC configuration is DOS or Windows 95/98 as the only system installed. The boot sequence for this case has already been outlined.

A similar concept can be used for Linux if it is the only operating system being used. In this case, one could theoretically skip the installation of LILO. The big disadvantage of doing this is that you can’t pass additional parameters to the system kernel at boot time.

As soon as there is more than one operating system installed, there are a number of new boot possibilities.

Booting another OS from a floppy disk: An OS can be booted from the hard disk. Other operating systems can be booted by using boot disks.

- *Requirements:* the floppy drive must be bootable.
- *Example:* install DOS or Windows 95/98 on the hard disk and boot Linux from a floppy disk.
- *Advantage:* you can skip the potentially tricky boot loader installation.
- *Disadvantage:* you have to *ensure* that you are not running out of working boot disks.

- It might be an advantage or disadvantage that your Linux is not capable of booting without a boot disk, depending on what it is to be used for.
- The boot process will take slightly longer.

Boot chaining of additional systems: The same OS is always booted and others can optionally be started from within the first OS.

- *Requirements:* adequate programs for chain booting of operating systems must be available.
- An *example* is the loading of Linux from DOS using loadlin or starting a NetWare server from DOS with server.exe.

Installing a boot manager: Theoretically, this allows you to use an arbitrary number of operating systems on a single machine. The choice of systems is done at boot time. Changing operating systems requires a reboot.

- *Requirements:* the boot manager must work smoothly with all installed operating systems.
- *Examples* of co-existing boot managers (at least under certain circumstances) are OS/2 and the DOS boot loader boot.sys.

The following section describes the installation and configuration of a boot manager, using the Linux boot manager LILO. A complete description of LILO's features can be found in [Alm94].

This file can be printed by entering

```
earth:/usr/share/doc/packages/lilo # lpr user.dvi
```

This is followed by a description of loadlin.

4.3 An Overview of LILO

The Linux boot loader is usually installed in the MBR (details below, page 97 and Section 4.5 page 104). When started, LILO already has access to both real mode hard disks and, due to its installation, is able to find all the data it needs from the *raw* hard drives without needing any information on partitioning. Because of this, operating systems can be booted from the first as well as from the second hard disk. The entries in the partition table that the standard DOS MBR uses to mark the active partition are ignored when using LILO in the MBR.

An important difference to the standard DOS boot sequence is that you can select any of the installed systems at boot time when using LILO. After loading the MBR into memory, LILO is started and you are asked to select one of the installed operating systems (see on the next page).

What is LILO?

LILO is a versatile boot manager. It can launch an operating system in the following ways:

- by loading the boot sector of a partition and starting an operating system from this partition. This is what other boot managers also do.

- by loading the Linux kernel and starting Linux. This cannot be done by most other boot managers.

Furthermore, LILO provides an important option of being able to pass a command line to the kernel. For security reasons, this can be protected totally, or partially, with a password.

How Do You Boot with LILO?

When LILO is launched, it displays the text `LILO` and a greeting message (which you entered during installation). Thereafter, a command prompt appears:

```
boot:
```

Here, you select your operating system by entering its name, which is then booted. The name of the operating system has previously been set during installation. At this point, you can pass a parameter line to the Linux kernel. You can also get a list of all the operating system names available by pressing TAB.

The Components of LILO

> **Tip**
>
> The boot sectors installed by LILO contain a typical virus boot sequence. DOS virus scanners typically claim to have found the *AIRCOP* boot sector virus in files such as `/boot/any_b.b` or `/boot/any_d.b`. Also, you should disable any BIOS protection of the MBR.

The LILO machinery consists of the following components:

- the beginning, or *first step*, of the LILO code in a boot sector which activates the system boot

- the *heart* of the LILO code, localized in `/boot/boot.b`

- a *map* file, normally `/boot/map`, where LILO enters the location of Linux kernels and other data during its installation

- optional: a *message file* whose contents are displayed as a welcome message before the LILO boot selection. Its usual location is:
  `/boot/message` (or similar)

- the different Linux kernel and boot sectors that LILO should offer

Caution

Any write access (even through file movements) on any of these files corrupts the map file, thus requiring you to *reinstall* LILO. This is only relevant when you change to another kernel.

Where LILO Can Be Installed

This is referring to the above-mentioned *first step* of LILO. Before going into detail, we would like to point out a very important general restriction:

Depending on the BIOS version on your computer, it may be required that the kernel image `/boot/vmlinuz` be located on the *first 1024 cylinders* of the hard disk. This can be achieved by a small extra partition which can be “mounted” in the directory `/boot`, and all of which is located within the first 1024 cylinders.

This is because these are the only cylinders available when the BIOS starts the system. With older BIOSes and IDE drives, you can also be restricted to either of the first two hard disks (`/dev/hda` and `/dev/hdb`).

If you have a very old BIOS, it is possible that the existence of (E)IDE hard disks could prevent your SCSI devices from booting.

Many newer BIOSes allow access to additional devices, for example, in connection with EIDE hard disk controllers for up to 4 EIDE devices. Many modern SCSI host adapters even allow SCSI devices to be “pushed to the front” in order to make them bootable. If you want to make use of this feature with LILO, have a look at the **`disk`** options on page 102.

For the sake of simplicity we will combine all this under the heading “1024 cylinder limit”. It should all be considered *before* a first-time installation (Section 2.4.1 page 35) — afterwards it may be too late, and may cause a lot of extra work. For more information, see 4.8.2 page 113.

The following locations are possible to store the LILO *boot sector*.

- **on a *floppy disk***. This is the most secure, but also the slowest alternative for booting with LILO (see Section 4.6 page 107). Choose this alternative if you do not want to change boot sectors.

- **in the *boot sector* of a primary Linux partition on the first hard disk**. This leaves the MBR untouched. Before it can be booted, the partition has to be marked active, with fdisk. If Linux is fully installed on logical drives or partitions on the second hard disk, there is only the boot sector of the extended drive of the first drive left (if there is one). Linux fdisk can also activate such a partition.

  If you want to boot multiple systems from hard disk, this is quite long-winded. Every time you want to boot, you have to activate the corresponding boot sector *beforehand*. The next two variants are much less cumbersome.

- **in the *Master Boot Record*.** This variation offers the highest flexibility. Moreover, this is the only alternative possible if all of the Linux partitions reside on the second hard disk and there is no extended partition on the first drive. Every setting of the MBR must be edited with extreme care since errors may have severe consequences. The safety aspects are described in Section 4.5 page 104.

- **If you have used *another boot manager* until now** and you want to continue using it. Depending on its flexibility and power, there are several variations. A common case: you have a primary Linux partition on the second hard disk where you boot Linux. Your boot manager is able to boot this partition via a boot sector. Then you can activate your Linux partition by installing LILO into this boot sector and telling your boot manager that it is active.

Caution

Be careful if you try to make a *logical* Linux partition bootable by installing LILO onto it. Success *is not guaranteed* at this point in time, even if your other boot manager is able to launch logical partitions.

Try it if you like. The safest way is to try it with a tiny Linux installation to see if it will work. Perhaps you will be lucky. The recommended way is still to create a primary and bootable Linux partition.

4.4 Configuring LILO

LILO is a flexible boot manager that offers many ways of adapting a configuration to one's needs. The most important options and meanings are described below. If you want to go into more detail, look at [Alm94].

Configuration of LILO is done in `/etc/lilo.conf`. If you are installing LILO for the first time, we recommend you use YaST to configure LILO. You can fine-tune, by editing `/etc/lilo.conf`, at a later stage.

Note

`/etc/lilo.conf` should only be readable for `'root'`, as it might contain passwords (see Section 4.4.2 page 101. This is the default setting with SuSE Linux. If in doubt, just check by invoking the following command as root:)

```
earth: # chmod 0600 /etc/lilo.conf
```

It is recommended you keep any existing old (and working) `lilo.conf` backed up in a safe place. Your settings only take effect when you reinstall LILO after changing `/etc/lilo.conf` (see Section 4.5 page 104).

4.4.1 Structure of `lilo.conf`

`/etc/lilo.conf` starts with a **global section** followed by one or more **system sections** for each operating system LILO should start. A new section is started by a line beginning with either **image** or **other**.

The order of entries in `/etc/lilo.conf` only matters in as much as the first one in the list is booted by default if no user interaction is taken —this can be set to **delay** and **timeout**.

A sample configuration for a machine with both DOS and Linux is shown in File contents 4.4.1. There are two Linux kernels (`/boot/vmlinuz` and `/boot/vmlinuz.old`) on `/dev/hda7`, as well as MS-DOS (or Windows 95/98) on `/dev/hda1`. The program MemTest86 is also alwailable.

```
### LILO global section
boot    = /dev/hda           # LILO installation target: MBR
backup  = /boot/MBR.hda.990428  # backup file for the old MBR
                                # 1999-04-28
vga     = normal             # normal text mode (80x25 chars)
read-only
menu-scheme = Wg:kw:Wg:Wg
lba32                        # Use BIOS to ignore
                             # 1024 cylinder limit
prompt
password = q99iwr4           # LILO password (example)
timeout = 80                 # Wait at prompt for 8 s before
                             # default is booted
message = /boot/message      # LILO's greeting

### LILO Linux section (default)
  image  = /boot/vmlinuz     # Default
  label  = linux
  root   = /dev/hda7         # Root partition for the kernel
  initrd = /boot/initrd

### LILO Linux section (fallback)
  image  = /boot/vmlinuz.suse
  label  = suse
  root   = /dev/hda7
  initrd = /boot/initrd.suse
  optional

### LILO other system section (DOS/Windows)
  other  = /dev/hda1         # Windows partition
  label  = windows

### LILO memtest section (memtest)
  image  = /boot/memtest.bin
  label  = memtest86
```

File contents 4.4.1: Sample configuration in `/etc/lilo.conf`

Anything between a `'#'` and the end of line is regarded as a comment. Spaces and comments are ignored by LILO and can be used to improve readability.

Now we go through the most important lines step by step:

- **Global section** (Parameter part)
  - **`boot=`**<**`bootdevice`**>
    The device on whose first sector LILO should be installed. <bootdevice> may be: a floppy disk drive (`/dev/fd0`), a partition (e. g., `/dev/hdb3`), or an entire disk (e. g., `/dev/hda`). The last means installing LILO in the MBR. Default: if this option is missing, LILO is installed on the current root partition.
  - **`lba32`**
    With this option you can ignore the 1024-cylinder-limit of LILO if your BIOS supports this.
  - **`prompt`**
    Forces the LILO prompt to be displayed. The default is: no prompt (compare with **`delay`** further down). This is recommended if LILO needs to manage more than one system. In addition, **`timeout`** should be set to guarantee an automatic reboot if nothing is entered at the prompt.
  - **`timeout=`**<**`tenth-seconds`**>
    Sets a timeout for the prompted option, thus enabling an automatic reboot if no entry occurs in the given time. <tenth-seconds> is the remaining time in 0.1 s. increments. Pressing `Shift ⇑` starts the timeout over. Default: infinite, e. g., no automatic reboot.
- **Linux section**
  - **`image=`**<**`kernelimage`**>
    Here the name of the kernel image to be booted, including its directory location, should be entered. With your new system, this is most probably `/boot/vmlinuz`, or `/vmlinuz` for older SuSE Linux systems.
  - **`label=`**<**`name`**>
    This name has to be unique in `/etc/lilo.conf`. Otherwise, you can freely choose a name for the system (e. g., **`Linux`**). Maximum length is 15 characters. You should use only letters, numbers and underscore for names – no blanks or special characters, for example. For more on the specific rules for which characters to use, see [Alm94], 3.2.1. The default is the filename of the kernel image (e. g., `/boot/vmlinuz`).

    By entering this name at the LILO prompt, you select which system to boot. It is recommended that, if there are many systems installed, that you keep track of them in a special message file (**`message`**).
  - **`root=`**<**`rootdevice`**>
    This is to give the kernel the name of the root partition (e. g., `/dev/hda2`) of your Linux system. This is recommended for security reasons. If this option is omitted, the kernel takes its own root partition <kernelimage>.

- **Linux part** (Fallback)

  Even if you installed a customized kernel, you are still able to boot the SuSE standard kernel to start a stable system.

  – **`optional`**

    If you decide to delete `/boot/vmlinuz.suse` (*not recommended*), this section will be skipped without an error message during LILO installation.

- **Other systems**

  – **`other=`<`partition`>**

    `other` tells LILO to start the partitions of other systems (e. g., `/dev/hda1`).

  – **`label=`<`name`>**

    Name (your own choice) for the system. Recommended, because the default—the raw device name—is less informative.

4.4.2 Other LILO configuration options

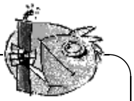
The previous section covered the entries required in `/etc/lilo.conf`. Other useful options are discussed below.

Those options that are marked as image options belong to the appropriate section of the operating system. The others are intended for the global parameter section of `/etc/lilo.conf`.

- **`backup=`<`backup`>**

  The file where LILO backs up the boot sector. The default is `/boot/boot.xxxx`, where `xxxx` is the internal device number of the installation partition.

  We do not recommend use of a cryptic name (see our example above). You will not be able to use the implemented uninstall feature of LILO, but we think it is better to this carefully by hand, anyway. (see Section 4.5 page 105)



  Caution

  If the backup file exists, LILO does *not* create a new one. Make sure you use a name not already in use.

- **`compact`**

  This option is recommended if you want to install LILO onto a floppy disk. If enabled, LILO tries to read more sectors at a time, resulting in a faster boot process. This does not work on every machine. We do not recommend that you set this as the normal way is safer and it only provides a difference of one or two seconds.

- **loader=<Boot loader>**

  To load a boot sector that belongs to another operating system, LILO constructs a *pseudo MBR* in its map file. At boot time, LILO first starts this pseudo MBR, which in turn starts the other boot sector. This option specifies the file where the code for the pseudo MBR is to be found.

  Default: `/boot/chain.b` (usually, this is correct).

  Sometimes another OS that needs to be booted from the first hard disk (e. g. DOS) is supposed to boot from another hard disk using LILO. There are additional options that cause the hard disks to swap according to their device numbers:

  map-drive=<Number> and **to=<Number>**. See: File contents 4.4.2 (on this page).

  The loader `os2_d.b` serves to load OS/2 from the second hard disk. [1] *New in LILO-Version 20:* “switching” devices has to be set explicitly now (see File contents 4.4.2)

```
# Booting DOS from the second hard disk
# DOS bootable partition config begins
other = /dev/hdb1
  label = DOS
  loader = /boot/chain.b
     map-drive = 0x80    #  first hd: BIOS number 0x80
     to        = 0x81    # second hd: BIOS number 0x81
     map-drive = 0x81
     to        = 0x80
  table = /dev/hdb
# DOS bootable partition config ends
```

File contents 4.4.2: `/etc/lilo.conf` Extract: Booting DOS from 2nd hard disk

- **table=<ptable>**

  <ptable> sets the source device for the partition table written into the pseudo MBR (normally `/dev/hda` or `/dev/sda`).

- **disk=<device file>**
  bios=<BIOS device number>
  cylinders=<amount>
  heads=<amount>
  sectors=<amount> Here you can tell LILO precisely which BIOS device number and geometry it should use. This is scarcely ever needed. There is one major exception: *IDE-SCSI* system: If you own a BIOS that is capable of switching the boot devices *SCSI prior to IDE* and you want to use this feature, you need to tell LILO the switched order from the perspective of the BIOS. This is achieved by an extra entry in the global section of `lilo.conf`.

[1] `any_b.b` (Booting from `B:`) and `any_d.b` (Booting from second hard drive) are obsolete from LILO-Version 20.

An example for a system with one SCSI and one IDE disk may be seen in File contents 4.4.3.

```
# Enable LILO to correctly access /dev/sda and /dev/hda
# at boot time if their boot order is interchanged in
# the BIOS:
disk = /dev/sda    #   The SCSI disk is regarded as ...
   bios = 0x80     #   ... first BIOS disk;
disk = /dev/hda    #   the IDE disk is regarded as ...
   bios = 0x81     #   ... second BIOS disk.
```

File contents 4.4.3: `lilo.conf` Extract: Boot order: SCSI prior to IDE

- **`linear`**

  Giving this option when installing LILO causes all references to hard disk sectors to be stored as logical instead of physical addresses, so that they are independent of any hard disk geometry. This option is intended for cases where, when booting, the BIOS detects a different geometry than that of the Linux system running. Only needed in rare cases!

  The **`linear`** option does *not* release you from the constraints of the 1024-cylinder-limit, which is determined by the BIOS geometry of the boot hard disk. Refer also to `file:/usr/share/doc/sdb/de/html/kgw_lilo_linear.html`.

- **`message=`**<**`message-file`**>

  Points to a text file that should be shown on screen at system boot up. It should not contain more than 24 lines and can present an overview of the LILO boot selection to augment the information available by pressing TAB. Recommended.

> Note
>
> If this option is set, the message file is then part of the LILO boot machinery and, after every change to this file, LILO has to be reinstalled (Section 4.5 on the next page).

- **`password=`**<**`password`**>

  May be located either in a global or system-specific section. Provides secure access to LILO services or booting the corresponding system by means of a password. If you take this seriously, you should remove the password from `lilo.conf` after you have used it for the first time. As `'root'`, you can set a new password for LILO any time you like (you just need to reinstall it afterwards) It is recommended to also set the option **`restricted`**, otherwise it could be possible to launch a shell, see manpage for **`lilo.conf`** (**`man lilo.conf`**)!

- **`read-only`**

This option tells the kernel to initially mount the root partition read-only, which is normal when starting Linux systems. If this is omitted, the kernel uses its internal settings.

- **`delay=`**<**`tenth-seconds`**>

  If the prompt is *not* explicitly set, you can order a prompt by pressing Shift ⇑, Ctrl, Alt. The **`delay=`** option sets the time to elapse before LILO boots the first system in its list. The default is 0, that is, no waiting.

  The **`delay`** option has no effect if a prompt is specifically requested by **`prompt`**.

- **`vga=`**<**`mode`**>

  Selects VGA mode at startup. Valid modes are `normal` (80x25), `ext` (80x50) or `ask` (ask at boot-time).

- **`append="`**<**`parameter`**>**`"`** Image option for Linux kernel. Enables kernel parameters and hardware components to be specified, in the same way that this is possible at the LILO prompt. The kernel first gets the **append** line, then the prompt. **`append="mcd=0x300,10"`**.

4.5 Installing and Uninstalling LILO

During a new Linux installation, or at a later time, YaST will lead you through the steps of how to install LILO interactively.

In this section, we assume that some action is required that goes beyond what YaST can accomplish, and we take a closer look at how LILO works during the installing and uninstalling process.

> **Caution**
>
> The installation of a boot manager is tricky! Ensure *in advance* that you are *100%* able to boot Linux and other mounted systems. You must have fdisk installed on a crash recovery disk, otherwise you might find yourself in the awkward situation of not being able to access your hard disk at all!

Installation After Changing the Configuration

If any of the LILO components have changed, or you have modified your configuration in `/etc/lilo.conf`, you will have to reinstall LILO. This is easily done by launching the "*Map Installer*" like this:

```
earth: # /sbin/lilo
```

What happens now is that LILO writes a backup of the target boot sector, writes its *first step* into it and creates a new map file (see also Section 4.3 page 96). LILO now announces each installed system—for an example see Output 4.5.1 on the next page.

```
Added linux *
Added suse
Added windows
Added memtest86
```

Output 4.5.1: Output after launching LILO

When the installation is complete, the machine can be rebooted:

```
earth: # shutdown -r now
```

During reboot, the BIOS first performs its system test and directly afterwards you will see LILO and its command prompt, where you can enter parameters and select a boot image from the recently installed configurations. TAB shows you a list of all systems installed.

Installation After Recompiling a Kernel

If you want to include a freshly created kernel into your LILO boot setup, the Linux kernel Makefile offers an all-in-one solution. All the commands to configure and create the kernel are put together in the file `/usr/src/linux/Makefile`; here the `INSTALL_PATH=/boot` is specified (see Section 9.5 page 241). This `Makefile` has a **`target`** called **`bzlilo`** which, after a kernel compilation, automatically copies the currently installed kernel `/boot/vmlinuz` (this used to be `/vmlinuz`) to `/boot/vmlinuz.old`, the new kernel to `/boot/vmlinuz`, and then reinstalls LILO. This can be done by entering the command:

```
earth:/usr/src/linux # make bzlilo
```

instead of make zImage. This is only useful if you have edited `/etc/lilo.conf` *in advance* and if your current kernel really is located in`/boot/vmlinuz`. The new, as well as the old, kernel should now be listed. See File contents 4.4.1 page 99 for an example of the resulting `/etc/lilo.conf`.

At the LILO prompt, you can launch either of the two kernels. This makes your boot more secure because you can still boot your old kernel even if the new one fails.

For more on creating a new kernel, see Chapter 9 page 237.

Uninstalling LILO

> **Caution**
>
> Uninstalling a boot manager is tricky! Please ensure *in advance* that you are *100%* able to boot Linux and other systems with their respective boot disks. You should have fdisk installed on every boot disk, otherwise you might find yourself in the unfortunate situation of not being able to access your hard disk at all!

Perhaps one day it will be necessary for you to uninstall LILO. This is accomplished by writing back the target boot sector where LILO has been installed. This is not a problem in Linux *if* there is a valid backup (see Section 4.4.2 page 101, option **`backup`**).

Caution

A boot sector backup is no longer valid if the partition in question has got a new file system (for DOS users: has been formatted). The partition table of an MBR backup becomes invalid if the hard disk in question has been repartitioned in the meantime. Obsolete "backups" are time-bombs. It is best to delete them as soon as possible. Unpacking old and invalid backups into system sectors is a direct route to data loss!

It is very simple to get back a DOS, Windows 95/98 or OS/2 MBR. Just enter the MS-DOS command (available since 5.0)

```
C:\> FDISK /MBR
```

or on OS/2

```
C:\> FDISK /NEWMBR
```

These commands only write the first 446 bytes (the boot code) into the MBR and leave partitions untouched.

For other restorations, first make a backup of the LILO sector in question—just to be on the safe side. Now you should check (at least twice) whether your old backup file is the correct one and if it is exactly 512 bytes in size! Finally, write it back (but do not confuse `if=` and `of=`)!

- If LILO resides in partition yyyy (e. g., hda1, hda2,. . .):

  ```
  earth: # dd if=/dev/yyyy of=New-File bs=512 count=1
  earth: # dd if=Backup-Date of=/dev/yyyy
  ```

- If LILO resides in the MBR of zzz (e. g., hda, sda):

  ```
  earth: # dd if=/dev/zzz of=New-File bs=512 count=1
  earth: # dd if=Backup-Date of=/dev/zzz bs=446 count=1
  ```

The last command is "cautious" and does not overwrite the partition table. Again, *do not forget:* with **`fdisk`** you should mark the desired starting partition as *bootable*. By the way, note how easy and fast a boot sector backup is done. We recommend you do this frequently!

4.6 Creating a Linux Boot Disk

A Linux boot disk consists (somewhat simplified) of one or more Linux kernels, possibly managed by LILO. It serves to start up your system even if it is not possible to boot directly from hard disk (possible reasons: overwritten MBR, misconfigured boot manager, errors while installing, etc.).

A boot disk such as this loads *only* the kernel. Everything else, including working system programs and `init` start scripts, must be provided by the installation on the hard disk. The connection between the boot disk and the system on the hard disk is established by the fact that in the kernel the root partition in question is set as the root device.

Do not confuse this with the SuSE boot disk which is used for installation and emergencies. If you need to create a new SuSE boot disk, you copy the appropriate image from the directory `disks` on the SuSE CD-ROM to a floppy disk (see Section 12.6 page 302).

Boot Disk Without LILO

Normally, if your hard disk controller support has been integrated into your kernel, your kernel does *not* need any additional command line with hardware info, etc. The easiest way to create a boot disk is to just write the actual kernel onto a raw disk and to adjust the root device (if this has not been done before).

```
earth: # /sbin/badblocks -v /dev/fd0 1440
earth: # dd if=Your_Kernel of=/dev/fd0 bs=18k
earth: # rdev /dev/fd0 Your_Root_Partition
earth: # rdev -R /dev/fd0 1
```

The last command makes sure that the kernel initially mounts root as read-only (the startup scripts expect this).

Boot Disk with LILO

You can create a much more capable boot disk with a greeting, prompt, kernel parameters and other LILO goodies, by transferring the complete LILO booting start machinery onto the disk (see Section 4.3 page 96). For this, the disk needs a file system; the Minix file system is best suited for this.

To do this, proceed as follows:

- Create a Minix file system on a new and empty floppy disk and mount the disk to, for example, `/mnt` with the commands:

  ```
  earth: # /sbin/mkfs.minix -c /dev/fd0 1440
  earth: # /bin/mount /dev/fd0 /mnt
  ```

- Now copy your kernel files and the LILO file `/boot/boot.b` to `/mnt`, for example, onto the floppy disk.
- Optional: create a message file `/mnt/message`.
- Create `lilo.conf` on `/mnt`. You need to adapt this to your needs (give the correct name of the kernel, etc.). See File contents 4.6.1 on the following page for an example.
- If you are using an `initrd`, don't forget to add it to `lilo.conf` – as described in 12.4.3 page 295.
- Install LILO with *this* `lilo.conf`:

  ```
  earth: # /sbin/lilo -C /mnt/lilo.conf
  ```

```
# LILO Configuration file bootdisk
# Start LILO global Section
boot=/dev/fd0                # Installation: Floppy
install=/mnt/boot.b          # Of course LILO and
map=/mnt/map                 # map file onto floppy!
message=/mnt/message         # optional
prompt
timeout=100     # Wait at prompt: 10 s
vga = normal    #
# End LILO global section
#
# Linux bootable partition config begins
image = /mnt/vmlinuz         #   default
    root = /dev/Your_Root_Device # Here is your root partition!
    label = linux
# Linux bootable partition config ends
#
# System sections for more kernel here:
```

File contents 4.6.1: `lilo.conf` for a bootdisk

- Unmount the floppy—that's it!

  ```
  earth: # /bin/umount /mnt
  ```

- Do not forget to check your boot disk at the next system start to check whether it works or not.

4.7 Sample Configurations

If Linux is the only operating system on your machine, there is nothing to do, since everything needed has already been done by YaST.

Now we will give you some example configurations for multi-system computers. Further information can be found under `/usr/share/doc/howto/en/mini/Linux+*.gz`.

4.7.1 DOS/Windows 95/98 and Linux

Requirements: There must be at least a primary partition for each of DOS/Windows 95/98 and Linux which is below the 1024 cylinders limit (Section 4.3 page 97).

For this case, we have already discussed a configuration (File contents 4.4.1 page 99) — only the settings for **`root`**, **`image`** and **`other`** have to be adapted. LILO is installed in the MBR.

Save your `/etc/lilo.conf` and be sure you have a Linux boot disk. Windows 95/98 feels especially inclined to eliminate "foreign" MBRs. If you can still boot Linux using your boot disk, this problem is quickly solved with the command

```
earth: # /sbin/lilo
```

which will complete your LILO installation.

4.7.2 Windows NT and Linux on One Hard Disk

1. If Windows NT and Linux need to co-exist on the same hard disk, you should use the NT boot manager for booting. This can either start the kernel images or the boot sectors themselves. Execution of the following steps prepares everything for a peaceful coexistence of Linux and Windows NT:

   - Install NT.
   - Partition the NT disks (using FAT so that Linux can write on it).
   - Install Linux as usual (in our example, the root partition is on `/dev/sda3`). Mount either the DOS partition or an error free DOS floppy disk (for example, on `/dos`).
   - Install LILO, but install it in Linux's root partition (`/dev/sda3`), **not** in the MBR (`/dev/sda`). You may still configure a selection of Linux kernels for LILO. See File contents 4.7.1 for an example `lilo.conf`.

```
# LILO Configuration file
# Start LILO global Section
boot=/dev/sda3                # Target of installation
backup=/boot/boot.sda3.970428 # Backup previous boot sector;
                              # 28. Apr 1997
prompt
timeout=100     # Wait at prompt: 10 s
vga = normal    # force sane video state
# End LILO global section
# Linux bootable partition config begins
image = /vmlinuz             #   default image to boot
    root = /dev/sda3         # Here the root partition!
    label = Linux
# Linux bootable partition config ends
```

File contents 4.7.1: `lilo.conf` for booting a Linux root partition

- Copy the LILO boot sector to a location where NT can find it; e. g.:

  ```
  earth: # /bin/dd if=/dev/sda3 bs=512 count=1 of=/dos/bootsek.lin
  ```

  This step as well as the following has to be performed after every kernel update.

- Boot NT. Copy `bootsek.lin` from data disk to main directory of NT's system drive (if it is not already there).
- In `boot.ini` (first setting attributes), supplement at the end:

  ```
  c:\bootsek.lin="Linux"
  ```

- After the next boot (if everything went smoothly), there should be an entry in NT's boot manager.

2. Another possibility: install LILO in the MBR and claim that it's DOS to Windows NT (as in our previous example).

> **Caution**
>
> NT 3.5* and 4.0 do not recognize Linux partition types 82 and 83. Make sure that no NT program tries to "repair" your partition table. This would result in loss of data! Always have valid backups of the LILO boot sector at hand.

4.7.3 OS/2 and Linux

1. Use the *OS/2 boot manager* for booting. It can launch unlimited primary and logical partitions provided they are below the 1024 cylinders limit. The user is responsible for the partitions. This boot manager is configured by OS/2's fdisk.

   Preparation on the Linux side: consists merely of making a partition bootable (usually this is the root partition) with LILO. You can use the same `lilo.conf` as in the Windows NT example, but there is one thing that you have to consider *in advance*. . .

   Preparation on the OS/2 side: OS/2 not only uses the conventional and obvious entries for existing partitions in MBRs on hard disks, but also uses "waste" space in these sectors for additional information. If these are inconsistent, OS/2's fdisk considers these partitions faulty and refuses to provide boot manager services. The fdisk commands of other systems do not know how to handle these extra partitions. Conflicts are inevitable.

   Therefore: before installing Linux, load OS/2 (the installation system is sufficient) and create the Linux partitions with OS/2's fdisk, at least the logical drives. This initially creates additional OS/2 partitions, which might get in the way.

   Solution: immediately after creating the partitions, load the Linux installation system (or the rescue disk from the SuSE Linux CD) and change the partition types to 83 (Linux native) using fdisk. Now these partitions will be ignored by OS/2.

2. *2nd option:* LILO is used as the main boot manager on a primary partition on the first hard disk. This is a special case in the following example, which also includes DOS. Using the MBR is less advantageous, since repartitioning with another fdisk could erase the MBR and thus remove LILO. This special case is also considered in our next example, where DOS is additionally involved.

4.7.4 DOS, OS/2 and Linux

1. If you have used the *OS/2 boot manager* for DOS and OS/2 and want to continue using it, simply add Linux to its start menu as described in the previous example.

2. If you have *LILO* installed as the main boot manager on a primary partition of the first hard disk, then the following, intentionally complicated example for `lilo.conf` (File contents 4.7.2) assumes that the DOS and Linux boot partitions are primary and on the first hard drive, whereas OS/2 resides on the second hard disk—all of them below the 1024 cylinders limit. OS/2 is on the second drive. This is why a special loader (`/boot/os2_b.b` instead of `/boot/chain.b`) is used. The MBR code might be either from DOS or OS/2 (it doesn't matter which). The LILO boot partition (`/dev/sda4`) must be marked as active, with any fdisk.

```
# LILO Configuration file
# Start LILO global Section
boot   = /dev/sda4        # LILO in Linux root partition
backup = /boot/boot.sda4.970428
message = /boot/message  #  Greeting message
prompt
delay = 100
vga   = normal
#
# Linux bootable partition config begins
image = /vmlinuz
    label = linux
    root  = /dev/sda4
# Linux bootable partition config ends
#
# OS/2 bootable partition config begins
other = /dev/sdb5
    table = /dev/sdb
    label = os2
    loader = /boot/os2_b.b
# OS/2 bootable partition config ends
#
# DOS bootable partition config begins
other = /dev/sda1
    table = /dev/sda
    label = dos
# DOS bootable partition config ends
```

File contents 4.7.2: LILO with DOS, OS/2 and Linux on two hard disks

4.8 LILO Problems

Some Guidelines

Some simple guidelines at the beginning will avoid most LILO problems in advance (this is taken from the LILO documentation [Alm94]):

- *Don't panic!* If anything does not work, try to find the error or the cause first; check the diagnosis before you start fixing the problem.
- Always have an up-to-date and tested *boot disk* at hand.
- SuSE Linux contains a full Linux system on its boot disk and installation CD (for the rescue system, see Section 12.6 page 302) to allow you to reach all your Linux partitions. Tools are included for repairing almost any problems that can occur.
- Read the complete LILO documentation, especially if the system does not do what you want it to do.
- Check `/etc/lilo.conf` *before* using the map installer (`/sbin/lilo`).
- Be careful if you are using a large hard disk, or multiple ones. You need to be aware of the 1024 cylinders limit.
- Try with and without the **`linear`** option (normally it should be better without!).

4.8.1 Diagnosis of Errors: LILO Start Messages

This is mainly section 5.2.1 from [Alm94].

When LILO loads itself, it displays the word `'LILO'`. Each letter is printed before or after performing some specific action. If LILO fails at some point, the letters printed so far can be used to identify the problem.

nothing No part of LILO has been loaded. Either LILO is not installed at all or the partition on which it's boot sector is located isn't active.

`'L'` *error* ... The *first stage* boot loader has been loaded and started, but it can't load the second stage boot loader (`/boot/boot.b`). The two-digit error codes indicate the type of problem. This condition usually indicates a media failure or a geometry mismatch.

`'LI'` The second stage has been invoked but could not be started. This can either be caused by a geometry mismatch or by moving `/boot/boot.b` without reinstalling LILO.

`'LIL'` The second stage of boot loader has been started, but it can't load the descriptor table from the map file. This is typically due to a physical error of the boot device or a faulty disk geometry.

`'LIL?'` The second stage boot loader has been loaded at an incorrect address. This is typically caused by a subtle geometry mismatch or by moving `/boot/boot.b` without reinstalling LILO.

`'LIL-'` The descriptor table (in the map file) is corrupt. This can either be caused by a geometry mismatch or by moving `/boot/boot.b` without reinstalling LILO.

`'LILO'` All parts of LILO have been successfully loaded.

The most common causes for *geometry errors* are not physical defects or invalid partition tables but errors in LILO installation, including:

- disregarding the 1024 cylinders limit (see next section)
- an unsuccessful attempt at starting LILO from a logical partition

4.8.2 The 1024-Cylinder Limit

> **Note**
>
> Recently BIOS versions are available which enable you to start operating systems above the 1024 cylinder limit. The current LILO version can use this BIOS extension. YaST and YaST2 will inform you accordingly of these options for your BIOS while configuring LILO. If your BIOS should not include this extension, you should definitely continue reading here.

As emphasized before (e.g. page 97), the entire LILO machinery (e. g., any data that is needed for booting) must be able to process BIOS calls (which means it must reside below the 1024 cylinders limit on the hard disk). The sections of the hard disk that can be used (these are called *allowed sections*) have already been discussed.

This restriction affects *only* the boot-up machinery. It is not required that LILO be installed on the Linux root partition. It is even possible (but quite dangerous, too) to put the boot machinery onto partitions of other operating systems to which Linux has read and write access.

> **Caution**
>
> Never install the LILO boot sector onto an unknown partition because you will severely damage the file system!

- The best method is to create a primary partition (within the allowed section) and to install all LILO files (including the LILO boot sector) into this partition. This will be, in most cases, the Linux root partition.

  You can also add it to `/boot` with YaST. The only condition is that there has to be enough space for

  - boot.b, map, message, and
  - the Linux kernels that LILO should boot.

  A few megabytes is enough. It does not matter where you put the rest of your partitions. There are no more restrictions. As soon as the kernel runs, you have unrestricted access to all installed drives.

But what to do if there is no space for such a partition? If you neither want to repartition your hard disk, upgrade to SCSI, or want to purchase a new BIOS version, there are still two (makeshift) possibilities:

- Use a boot disk instead of LILO on the hard disk or, if you are also running MS-DOS, you can use loadlin as well.

- Install the LILO boot machinery onto a Linux partition which is in the permitted section and where Linux has write access (e. g., a FAT/VFAT drive). We cannot put the LILO boot sector there as well! So there are only two places to put it. Either at the start of an extended partition on the first drive—as long as it is beneath the 1024 cylinders limit—or on the MBR.

  Suppose that the partition in question is mounted on /mnt, that LILO is installed in the MBR (/dev/hda), and that you also boot DOS from /dev/hda1. Then you should proceed as follows:

  - Create a new directory (e. g., /mnt/LINUX) and copy the LILO files mentioned above to it: boot.b, map, message, as well as the chain loader of other operating systems (normally chain.b) and finally the Linux kernels that LILO should boot.
  - Create a /mnt/LINUX/lilo.conf where all paths point to /mnt/LINUX (see File contents 4.8.1 on the facing page).
  - Install LILO with *this* lilo.conf:

    ```
    earth:  # /sbin/lilo -C /mnt/LINUX/lilo.conf
    ```

    After that, LILO should work. Boot MS-DOS and protect the LILO files as well as possible against write access (just to remind you: any write access disables LILO). To accomplish this, you should assign to all files in X:\LINUX (where the 'X' is the DOS drive mounted to /mnt) the DOS attributes *system* and *hide*.

In conclusion, we point you toward two HOWTOs in /usr/share/doc/howto/en/mini/—LILO.gz and Large-Disk.gz.

4.8.3 Special Boot Problems with Kernels from 2.0 Onwards

Problems

Problems booting with LILO might occur after part of the kernel has been loaded. (e. g., a SuSE installation kernel)

You can select a kernel at the LILO prompt and this kernel is loaded (some dots are output onto the screen) but starting the kernel fails. Before reaching "uncompressing Linux" the system crashes with different behaviors.

Possible error messages:

```
# LILO Configuration file
# Start LILO global Section
boot=/dev/hda              # Installation target
backup=/mnt/LINUX/hda.xxxx  # backup of old MBR
install=/mnt/LINUX/boot.b  # Of course LILO and
map=/mnt/LINUX/map         # map file are in /mnt/LINUX!
message=/mnt/LINUX/message # optional
prompt
timeout=100     # Wait at prompt: 10 s
vga = normal    #
# End LILO global section
#
# Linux bootable partition config begins
image = /mnt/LINUX/First_Kernel   #   default
    root = /dev/Your_Root_Device    # Root partition!
    label = linux
# Linux bootable partition config ends
#
# System section for other kernels:
#
# End Linux
# DOS bootable partition config begins
other = /dev/hda1      # MSDOS system drive
    label = dos
    loader = /mnt/LINUX/chain.b
    table = /dev/hda
# DOS bootable partition config ends
```

File contents 4.8.1: `lilo.conf` for other partitions

- System reboots
- System just hangs
- `"crc-error"`
- `"no free space"`
- `"Error 0x00"`
- `"Error 0x01"`
- `"incomplete literal tree"`

Thereafter, access on the floppy is attempted but the system hangs.

Cause

The cause lies in a combination of a big kernel, LILO and faulty hardware. This affects roughly 1% of all machines.

We assume that this is due to a faulty BIOS that has problems with fast memory access.

This problem does *not* occur, if:

- the machine is booted via loadlin
- the kernel has been copied to a floppy

  ```
  dd if=/vmlinuz of=/dev/fd0
  ```

  and booted from there
- a smaller kernel is used, which has been created with

  ```
  make zImage
  ```

  (e. g., an older 1.2.13 kernel) is booted via LILO

The following BIOS settings do not cause any problems either:

- Disable Internal Cache
- **`DRAM Precharge Wait State`** value: 1 *and*
- **`DRAM Wait Burst Timing`** value: `0x3333`

Solution

First of all, you should be able to install a system. So, if you can neither boot via loadlin nor via setup, you should use an old kernel 1.2.13 boot disk for installation. If you do not have such a disk at hand, you should change the BIOS settings accordingly.

After a successful installation, the question is how you want to boot in the future. First, you should use the same media as during installation. loadlin from DOS should not cause any problems. With a boot disk, you should enter the following parameters:

```
load_ramdisk=0 root=/dev/???
```

where ??? is your root partition (e. g., /dev/hda1). Thereafter, you should build your own kernel, since this can be booted using LILO.

4.9 Starting via loadlin

Now we want to offer an alternative to boot SuSE Linux, loadlin. The program loadlin is a DOS program that is capable of booting a Linux kernel from a DOS directory. Thus loadlin perfectly integrates itself into an existing DOS/Windows 9x environment. As no entry in the MBR is needed, Windows only notices one or more partitions with unknown ID's. The risk of unwanted side effects due to a Linux installation is thus minimized.

The procedure described below works on both Windows 95 and Windows 98. The files themselves have been written in Windows 95; for this reason we will just talk about Windows 95.

In principle, there are two ways of activating loadlin. One is to switch between various systems via a boot menu, or to start Linux via loadlin from another running system.

Both methods have advantages and disadvantages:

- A boot menu saves you the trouble of having to start another operating system first, before you can start Linux.
- You can add other configurations to your boot menu to create a universal starting mechanism.
- You need to modify start files, however, to build a boot menu; which you will have to do by trial and error.
- Changing to Linux from the DOS prompt is very simple.
- A Linux start can be very nicely integrated into a Windows 95 session. Double clicking an icon will start Linux (Windows 95 contains DOS 7.0).

Tip

You should use a boot menu if you are using DOS or Windows 3.x. If using Windows 95, you can start most easily from the running system. Start menus in Windows 95 are a very complex matter. We are only able to give you some hints about them.

4.9.1 Necessary Steps for all loadlin Users

This is what you need to do, whether you decide to use a boot menu, or to start from a running system (in DOS, Windows 3.x, or Windows 95):

1. You may have already installed loadlin. If not, do so now using setup.
2. Change to `c:\loadlin` in MS-DOS. There you will find the file `linux.par`. Create a file with the editor named `startlin.bat` (you can give it another name) in this directory. Now insert the line described in File contents 4.9.1:

```
c:\loadlin\loadlin @c:\loadlin\linux.par
```

File contents 4.9.1: Example of a batch file for starting Linux

Next, edit the file `linux.par` (File contents 4.9.2).

```
c:\loadlin\vmlinuz  # first value must be
                    # the filename of the Linux kernel

root=/dev/xxx       # the device which gets mounted as root FS

ro                  # mount root read-only
```

File contents 4.9.2: Example of the file `linux.par` with customized values

Instead of `xxx`, enter your root partition's device name (you wrote down this name in Section 3.3.12 page 66). `startlin.bat` starts Linux. The file `linux.par` is used by `startlin.bat` as well as by `config.sys`, and contains essential parameters. Later on, when you are more familiar with Linux, you can add or replace parameters here. If you have built a kernel of your own, just copy it to `c:\loadlin\vmlinuz` and from then on this kernel will be booted.

4.9.2 Setting up Boot Menus

Here's how to configure a boot menu in DOS or Windows 3.x:

1. First, define a boot menu section in your `c:\config.sys` file.
   Open `c:\config.sys` in an editor and enter something similar to the File contents 4.9.3 on the facing page.

   Under the label `[Menu]`, define an entry in the boot menu for each OS you want to boot. Also, define the menu's color and after how many seconds each OS will be automatically started.
2. Below these entries, enter the labels [Common], [Win], [DOS], and [Linux]. Commands entered in [Common] are always executed. All other entries are OS specific. See the example in File contents 4.9.4 on the next page. As a

```
[Menu]
menuitem=Win, starting Windows,...
menuitem=DOS, starting DOS, MS-DOS...
menuitem=Linux, starting Linux...
menucolor=15,1
menudefault=Win,5
```

File contents 4.9.3: Example of first part of Linux boot menu in `c:\config.sys`

```
[Common]
device=c:\dos\himem.sys /testmem:off
device=c:\dos\emm386.exe noems I=E000-F4FF
dos=high,umb
files=30
buffers=10
shell=c:\dos\command.com

[Win]
devicehigh=c:\dos\dblspace.sys /move
devicehigh=c:\cd\slcd.sys /D:SONY_000 /B:340 /M:P /V /C

[DOS]
devicehigh=c:\dos\dblspace.sys /move
devicehigh=c:\cd\slcd.sys /D:SONY_000 /B:340 /M:P /V /C

[Linux]
shell=c:\loadlin\loadlin.exe @c:\loadlin\linux.par

[Common]
rem Remains blank
```

File contents 4.9.4: Example of second part of Linux boot menu in `c:\config.sys` guide, you can use the lines in your own `config.sys`. An example may be found in: File contents 4.9.4 on the preceding page.

Now save the file with your changes.

3. Next, edit `c:\autoexec.bat`. Here you must put the same labels and assign entries to labels, but the notation differs slightly. See the example in File contents 4.9.5. Notice that the Linux case is not mentioned here because Linux is booted using loadlin directly from the `c:\config.sys` file. The variable %config% contains the selected label (:Win or :DOS). Customize the code to fit your machine.

```
@echo off

rem Entries for all Configurations
switches= /f
set comspec=c:\dos\command.com
prompt $p$g
loadhigh c:\dos\keyb gr,,c:\dos\keyboard.sys
loadhigh c:\dos\doskey
set temp=c:\temp
loadhigh c:\dos\mscdex.exe /D:SONY_000 /E /V /L:H
c:\logimaus\mouse.exe

goto %config%

:Win
c:\dos\smartdrv.exe a- b- c+ 2048 1024
path c:.;d:.;c:\windows;c:\dos;c:\util;
win :
c:\dos\smartdrv /C
goto ende

:DOS
path c:.;d:.;c:\dos;c:\util;
goto ende

:ende
echo * Goodbye *
```

File contents 4.9.5: Example of `autoexec.bat` supporting Linux boot menu

4. If you now boot your machine, the boot menu appears and you have five seconds to choose an operating system. Then Windows starts automatically (menudefault=Win,5). If you select ‘`Linux`’, Linux starts and awaits your login.

4.9.3 Starting Linux from Within Windows

To create a start icon for Linux to boot Linux from within a running Windows 95 session:

1. Open Windows Explorer. Change to `c:\loadlin`. With the right mouse button, click on the file `startlin.bat` and select ‘`Create Shortcut`’.
2. Drag the shortcut onto the desktop.
3. Click on ‘`Shortcut to startlin.bat`’ with the right mouse button and select ‘`Properties`’. Go to tab ‘`Program`’, click on the button ‘`Advanced`’ and click ‘`MS-DOS mode`’ on. Confirm with ‘`OK`’.
4. Click the ‘`Change icon`’ button and select a nice icon; give the shortcut a suitable name; Voila!
5. Double clicking this new shortcut should bring up a dialog box telling you that Windows 95 is about to switch to DOS mode. If this dialog box bothers you, turn it off in the properties menu.

4.9.4 The Windows Boot Menu

This is how you install a boot menu for Windows 95:

1. You must edit the file `c:\msdos.sys`. First, make the file visible by entering: `C:>` **`attrib -R -S -H c:\msdos.sys`**

   This is a text file where you have to enter some lines to deactivate the Windows 95 start menu. The [Options] label should resemble File contents 4.9.6.

```
[Options]
BootGUI=0
BootDelay=0
BootMenu=0
Logo=0
```

File contents 4.9.6: `msdos.sys` to start Linux using a Windows 95 boot menu

   The parameter `Logo=0` is optional and avoids switching to graphics mode before Windows 95 is started. Booting is much faster and you avoid lots of trouble if you plan to use the DOS emulator in Linux later on.

   The parameter `BootGUI=0` is for booting Windows 95 directly into DOS mode. To start Windows from this you have to enter:

   `C:>` **`win`** but this is already done by our example `c:\autoexec.bat` if you have selected Win95 from our menu.

2. Next, define your boot menu in `c:\config.sys`. See File contents 4.9.7 on the next page) for an example.

   Under the Label `[Menu]`, define entries for the boot menu, its color and the delay in starting up automatically.

3. Further down in `config.sys`, enter labels `[Win95]`, `[DOS]`, `[Linux]`, and `[Common]`. `[Common]` is for entries that apply every time (this will very rarely be the case in Windows 95). All other labels are for the corresponding operating system. Use those lines that are already written in your

```
[Menu]
menuitem=Win95, start Windows 95...
menuitem=DOS, start MS-DOS...
menuitem=Linux, start Linux...
menudefault=Win95,5
```

File contents 4.9.7: Example `config.sys` (part one) for starting Linux using a boot menu with Windows 95

`config.sys`. The example in File contents 4.9.8, should only be regarded as a hint.

```
[Win95]
dos=high,umb
device=c:\windows\himem.sys /testmem:off

[DOS] device=c:\plugplay\drivers\dos\dwcfgmg.sys
dos=high,umb
device=c:\windows\himem.sys /testmem:off
device=c:\windows\emm386.exe noems I=B000-B7FF
devicehigh=c:\cdrom\torisan.sys /D:TSYCD3 /P:SM

[Linux]
shell=c:\loadlin\loadlin.exe @c:\loadlin\linux.par

[Common]
accdate=C+ D+ H+
switches= /F buffers=20
```

File contents 4.9.8: Example `config.sys` (part two) for starting Linux using a boot menu with Windows 95

Now save this file.

4. Next, edit `c:\autoexec.bat`. Here, the same labels are entered, but the notation is slightly different. The label which has been selected is written to the variable %config%. Notice that the Linux case is not included here, because Linux is booted using loadlin directly from the `config.sys` file. Your entry should look something like File contents 4.9.9 on the next page.

   When you have finished editing the file, don't forget to save it.

5. If you now boot your machine, the Windows 95 boot menu should appear, giving you two seconds to select an entry. If you choose *Command line*, your own boot menu will be displayed. Now you have five seconds to select an operating system. After this delay, Windows 95 starts automatically. If you select '`Linux`', Linux starts and awaits your login.

```
@echo off
loadhigh keyb gr,,c:\windows\command\keyboard.sys
goto %config%

:Win95
win
goto ende

:DOS
path c:.;d:.;c:\windows\command;c:\util;
loadhigh c:\windows\command\mscdex.exe /D:TSYCD3 /L:x
loadhigh c:\windows\command\doskey
c:\windows\command\mouse.exe
goto ende

:ende
echo * And now? *
```

File contents 4.9.9: Example `autoexec.bat` for starting Linux in Windows 95

5 The X Window System

5.1 Historical Background

The X Window System is the de facto standard GUI for UNIX. Yet the X Window System is far more than this – X11 is a network-based system. Applications running on the machine `earth` can display their results on the machine `helios`, provided the two machines are connected via a network. The network could be a local one (LAN) or a connection between computers thousands of miles away via the Internet.

X11 was first developed as an enterprise of DEC (Digital Equipment Corporation) and the project Athena at MIT (Massachusetts Institute of Technology). The first release of X11R1 was in September 1987. Since release 6, the X Consortium, Inc. has been responsible for the development of the X Window System.

XFree86 ™ is a freely available implementation of X servers for PC systems. It was developed by a handful of ambitious programmers who founded the XFree86 team in 1992. In 1994, this team went on to found The XFree86 Project, whose aim is to continue research and development on X11 and to provide it to the public. Since March 2000 the completely revised major release XFree86-4.0 has been available for download from `http://www.XFree86.org`. By default, SuSE Linux installs XFree86-4.0. Below we will take a closer look at the features of the new version.

SuSE would like to thank the XFree86 team for their help and for their permission to include beta servers on our CD's[1], without which their production would have been much more difficult, if at all possible.

The next sections are about configuring the X server. For this purpose SaX2 [2] and xf86config will be discussed, simple tools for configuring the X Window System. SaX2 is intended for configuring XFree86-4.0, SaX is used to configure the XFree86 3.3.x versions.

In contrast to the text-based xf86config, the two SaX versions work directly with the X-server and they can be operated with the mouse. So install, using YaST, the program SaX (package `sax`, series `x`) or SaX2 (package `sax2`, series `x`), together with their dependent packages. Whereas XFree86-4.0 contains all the necessary graphics drivers in the base package, if you use XFree86-3.3.x, especially the package `xvga16` – and if it is already known – the X server which matches your graphics card must be installed; the X servers are listed in the series `xsrv` (see

[1] Parts of this documentation are taken from chapter *XFree86 Konfigurieren* from [HHMK96] which was kindly given to us by DIRK HOHNDEL

[2] SaX: ***S**uSE **A**dvanced **X** Configuration Tool* The configuration program SaX2 (**`sax2`**) to configure XFree86-4.0 makes XF86Setup (package `xfsetup`, series `x`) obsolete.

Section 3.4.3 page 72). If you have forgotten to select a server, SaX will detect this and install a suitable server, by means of YaST. In XFree86-4.0 all necessary graphics drivers are already included in the base selection.

In order to make optimal use of the hardware available (graphics card, monitor, keyboard), you can optimize the configuration manually. Certain aspects of this optimization will be explained. Even more detailed information on configuring the X Window System can be found in the directory `/usr/share/doc/packages/xf86` as well as in the manpage for **`XF86Config`** (**`man XF86Config`**).

Caution

Be very careful when configuring your X Window System! Never start the X Window System until the configuration is finished. A wrongly configured system can cause irreparable damage to your hardware (this applies especially to fixed-frequency monitors). The authors of this book and SuSE cannot be held responsible for damage. This information has been carefully researched, but this does not guarantee that all methods presented here are correct and will not damage your hardware.

5.2 The New Version 4.0 of XFree86

This version of SuSE Linux contains the current version, 4.0, of XFree86 which varies in a number of ways from version 3.3, used until now. Overall there are hardly any differences for the user when operating the graphical desktop; applications such as the graphical desktop KDE or GNOME behave with the new version in the same way as the version 3.3.6 used until now.

What advantages does the new version provide?

The new X server is no longer a monolithic program, but just a relatively small basic scaffolding to which the necessary program modules can be later added, if and when required. For example there are no longer many different X servers for different graphics cards, as in the previous version, but just one executable program called `XFree86`, which can be found in the directory `/usr/X11R6/bin`. This is also the actual X server. The graphics driver, which then takes on the task of controlling the graphics card, is a loadable module.

A similar method is used to support the various input devices, fonts or X protocols. This again consists of individual modules which can be later loaded by the X server. As a rule you don't need to worry about these modules, the configuration of the modules to operate the graphical desktop on your computer is managed as far as possible by SaX2.

Through this module concept, it is easy for a vendor to implement a driver for exotic hardware, such as touchscreens or brand-new graphics cards. The developers have even ensured that the necessary modules for various operating systems only need to be made available once, which means that a graphics driver module which was compiled in FreeBSD, for example, can also be used in Linux, and vice-versa. This portability, however, is of course limited to the same hardware platform: a module which was compiled for Linux on Power PC's cannot be used on an Intel PC.

Support for the mouse has also been significantly improved. Especially under heavy loads, the reaction of the mouse to mouse movements is considerably faster and more direct than with the previous XFree86 X server. Overall the output speed has also been improved, so that graphics operations are generally performed more quickly than on the old X server due to the completely revised XAA (*XFree86 Acceleration Architecture*).

Compared to XFree86 3.3.x, the configuration file has a slightly different format and is now located in `/etc/X11/XF86Config`: if you want to 'fine tune' your X configuration, details on the structure of the configuration file, and how it functions, can be found in section 5.6 page 162.

Error logging has also been improved. The X server creates a very detailed log file, which you can always find after the X server has started in the file `/var/log/XFree86.0.log`.

One of the further features of the new version is the support of special options such as True Type fonts. Other features also include the provision of the 3D protocol extension, **`glx`**, gamma correction of the screen and the support of multiple graphics cards for **`Multihead configurations`**. More information on this can be found in Section 5.6 page 162.

What has changed?

XFree86-4.0 is, of course, based on the version 3.3.x. Unfortunately not all drivers are included, as some are very complex and could not be ported to the new XAA architecture. If these graphics cards were supported by SuSE Linux before, you can still use these with the XFree86 version-3.3.x. For such cards, please use the XFree86 version 3.3.6, also contained in SuSE Linux. You can continue to use SaX to configure these cards, as before.

The graphics cards concerned here are those which were until now used together with the following X servers: XF86_S3, XF86_Mach8, XF86_Mach32 and XF86_8514.

In the case of S3 cards, those which previously required the S3 server are not supported by XFree86-4.0, while those which previously worked with the SVGA server will work correctly with XFree86-4.0.

Basically these are the graphics cards S3 Trio3D, Savage4, Savage3D and Savage2000 chips, and almost all S3 Virge cards.

Graphics cards which needed the other X servers listed above (Mach8, Mach32 and 8514) are no longer very common. For these graphics cards – as for all the old S3 cards – the XFree86-3.3 version continues to be available.

5.3 Configuration Using SaX2

The program SaX2 (***S**uSE **A**dvanced **X** Configuration Tool*) provides a simple installation of the X Window System. This is a successor to the tried-and-tested SaX program. SaX was used to configure XFree86-3.3.x, SaX2 is now used for the configuration of XFree86-4.0.

If your system is already set up with the X server of XFree86-4.0, then you can start SaX2 straight away. If you are using an XFree86-3.3.x X server, then a message will be displayed when starting SaX2 telling you that the system is set up with XFree86-3.3.x, meaning that SaX needs to be used for the configuration. If you still continue, XFree86-4.0 will be set up. Strictly speaking, this is not entirely correct: the base system, that is the libraries and the X11 base programs, are *always* taken from XFree86-4.0, merely the X server and the programs xinit and xf86config are replaced.

> Note
>
> As already mentioned in Section 5.2 page 126, not all graphics cards are supported by XFree86-4.0. To configure these graphics cards you should use SaX. This program is described in Section 5.4 page 146.

You can use the mouse or the keyboard directly in the graphical desktop in SaX2. Apart from a few special cases, such as very modern or very old hardware, it can detect the components used on its own, and configure them to a large extent automatically, so that installing an X server is a simple affair.

5.3.1 First-time installation

For the very first installation of the X Window System – the graphical user interface of every Linux system – a few details about the computer must be known:

- The monitor model and technical specifications (if known).
- The keyboard type.
- The mouse type and the port to which it is connected.
- The name and vendor of the graphics card.

> Note
>
> You have to start the program SaX2 (**sax2**) as the user 'root'. You can also start SaX2 from YaST: 'System administration' and then with 'Configure XFree86[tm]' (see Section 3.6 page 78).

At the command line the program is started with:

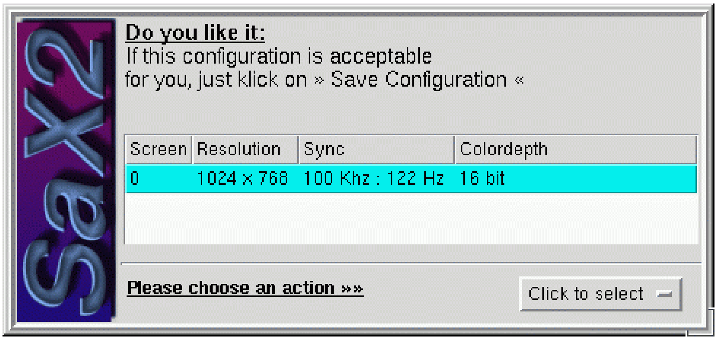
```
earth:/root # sax2
```

As soon as the program is started, a utility belonging to SaX2 called sysp searches for installed PCI and AGP graphics cards. It also determines the keyboard and mouse types. A configuration file is then put together from the data detected, which is used to start the X server. If, for example, an AGP or PCI graphics card was found in the computer, this is identified and the matching X server module is immediately loaded.

SaX2 remembers the settings it finds and stores them in a file. To force a completely new hardware detection, you can start SaX2 with the option **-f**. Then all the hardware relevant to X11 is freshly examined.

Automatic configuration

If the graphics card can be uniquely identified with a PCI scan and a graphics driver is available, SaX2 then tries to detect the monitor automatically via DDC ("Direct Data Connection"). If this is successful, the optimal resolution for the monitor is set. This is shown to you in the box 'Do you like it' and you can apply these by clicking on the 'OK' button. When you do this, make sure that in the selection box next to the 'OK' button 'Save' is also selected.



Do you like it:
If this configuration is acceptable
for you, just klick on » Save Configuration «

Screen	Resolution	Sync	Colordepth
0	1024 x 768	100 Khz : 122 Hz	16 bit

Please choose an action »»

Figure 5.1: SaX2: Fully automatic configuration

In the selection box next to the 'OK' button you can select further items. Click on 'OK' after you have selected the item you require.

- Via the item 'Adjust', you can alter the appearance of the image on your screen within certain limits, before you save the configuration.
- SaX2 does try and identify your hardware as far as possible, but to be absolutely sure that the configuration is complete and correct, you should always check all the settings of SaX2 and change them if necessary. You can, via the item 'Run SaX', start SaX2 again and change the settings of the X server there. This is particularly important if you have more than one graphics card in your computer or if the mouse does not work correctly at first.
- 'Exit Sax' closes down the program sax2.

In case your mouse is not correctly configured, you can also use the program with the keyboard. By repeatedly pressing the Tab key you can activate the individual entry fields of an index tab. To reach the various index tabs, just keep on pressing Tab until the heading of the current tab card is highlighted with a black frame; you can select the required index tab using the cursor keys, ← or →. After confirming your choice with ↵ (= Enter) this index tab is then displayed. Alternatively in SaX2 you can use a "virtual" mouse. To do this, hold down the Shift ⇑ key and use the L key to move the mouse cursor to the right, J to the left, K to move it down and I to move it up. Shift ⇑ and A produces a "click" with the left mouse button, Shift ⇑ and S one with the middle button, and Shift ⇑ and D, a "click" with the right mouse button.

To select an entry in a list box, press the Tab key again until the relevant box is highlighted. With ↑ and ↓ you can look for an entry through the colored highlighting, which is then set by pressing Enter, i.e. becomes active.

Tip

SaX2 contains its own help system which can be reached by pressing the keys Ctrl + F1. Here all the key combinations and mouse functions of SaX2 are listed.

By pressing the middle mouse button you will reach a special menu via which you can, among other things, take a closer look at the error log file of the X server. This can be extremely helpful in finding errors. An explanation of the protocol file of the X server, which is stored in the file `/var/log/XFree86.0.log`, can be found in Section 5.3.1 page 143.

...and manual configuration

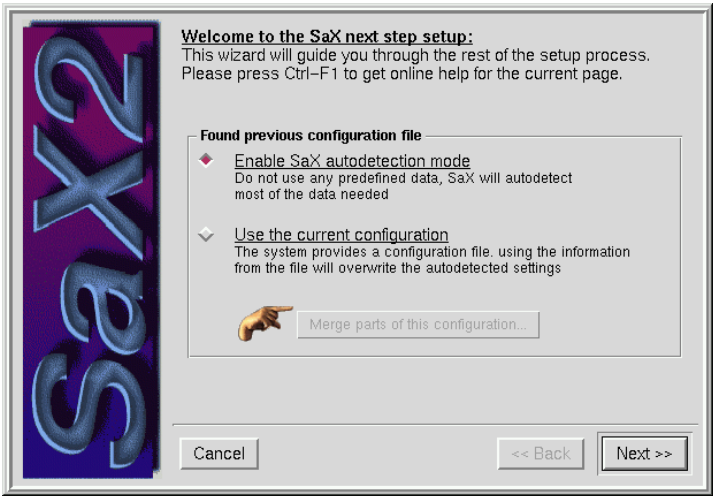
After selecting the item '`Run SaX`', the welcome dialog of SaX2 appears. After a few moments whilst SaX2 reads in a few data files, the program is ready. You can choose from two index tabs, '`Configuration`' und '`Expert`'. You also have the option of reading in a configuration file which may have been created in an earlier session.

Note

Via this item you can only read XFree86-4.0 configuration files. Old XFree86-3.3.x configuration files cannot be read here.

You can influence the further behavior of SaX2 via the items '`Custom`' or '`Easy`'. If you select '`Custom`', you can, in addition to the settings offered in '`Easy`', also change the paths for fonts, mouse and keyboard manually. In '`Easy`' these settings will be made automatically, or taken from the already configured system.

If you have chosen the item '`Easy`', clicking on '`Next`' will take you directly to the dialog configuring the graphics driver and the menus to configure mouse, keyboard and paths are omitted. The parameters needed to configure mouse and keyboard will then be taken from the automatic hardware detection, which was



Welcome to the SaX next step setup:
This wizard will guide you through the rest of the setup process.
Please press Ctrl–F1 to get online help for the current page.

Found previous configuration file

◆ Enable SaX autodetection mode
Do not use any predefined data, SaX will autodetect most of the data needed

◇ Use the current configuration
The system provides a configuration file. using the information from the file will overwrite the autodetected settings

Figure 5.2: SaX2: Welcome screen

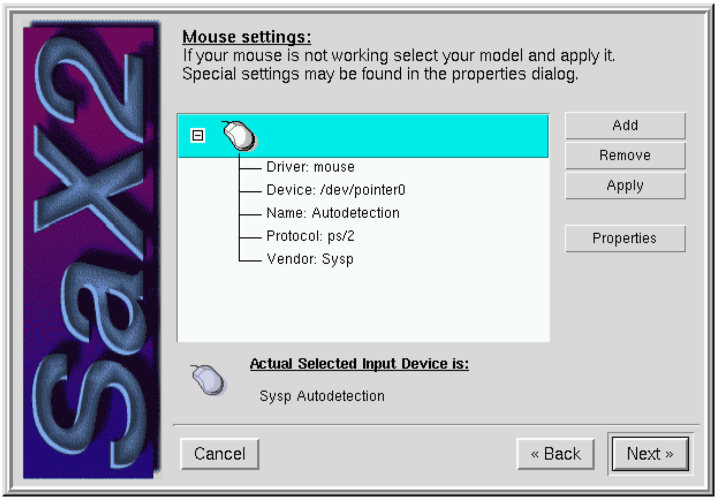
run when SaX2 was started. For language settings the contents of the environment variable, **LANG**, are used. For the search paths a sensible default is chosen. In Section 5.6.1 page 166 there is a description of how you can adjust the font paths in your system. This is of special interest if you want to process foreign language texts on your system.

Below is a description of the individual menu items if you have chosen the item ‘Complex’.

Mouse

After clicking on ‘Next’, the mouse configuration dialog will appear. If the mouse does not seem to function correctly, you can, of course, also use the keyboard in this dialog, see Section 5.3 on the following page. To do this, click repeatedly on the Tab key until the field ‘Mouse Tree’ is highlighted, and thus activated. Press Enter and select ‘Vendor’ and ‘Name’, the vendor and mouse type, and confirm your choice with ‘OK’. With the ↓ and ↑ keys you can select both a vendor and mouse type and confirm this with the Enter key.

If your mouse type is not listed here, you can also configure it manually. To do this return to the dialog ‘Mouse settings’ and activate the selection box ‘Protocol’ in the field ‘Mouse Properties’. At this point you can choose between activating the entry field yourself, or activating the selection list (the small symbol to the right of the entry field). Press the space bar to call up the selection list, or you can alternatively enter the X11 mouse protocol name in the entry field. You could also use the item ‘Device’; here you need to select the device file through which the connected mouse is reachable. If you have already set up the mouse with YaST2, just select /dev/mouse, otherwise you can look for the matching interface. Sensible settings are suggested in the selection box — **/dev/psaux** for PS/2 mice, **/dev/ttySx** for serial ones and **/dev/usbmouse**



Mouse settings:
If your mouse is not working select your model and apply it.
Special settings may be found in the properties dialog.

- Driver: mouse
- Device: /dev/pointer0
- Name: Autodetection
- Protocol: ps/2
- Vendor: Sysp

Actual Selected Input Device is:
Sysp Autodetection

Figure 5.3: SaX2: mouse configuration

for USB mice. Refer to table 5.1 for an overview of the various mouse types and their configurations in the X Window System.

Protocol used	mouse type
PS/2	2 or 3 button mouse on the USB port or on the PS/2 mouse connection.
IMPS/2	Wheel mouse with 3 or more buttons and one or more scroll wheels, connected to the USB port or the PS/2 mouse connection.
Microsoft	2 and sometimes 3 button mice, connected to the serial port.
MouseSystems	3 button mice on the serial port.
Intellimouse	Wheel mouse with 3 or more buttons and one or more scroll wheels connected to the serial port.
Auto	Automatic detection of the serial mouse connected.

Table 5.1: Different mice and the protocols used

For special settings you should take a look at the ‘`Expert`’ menu. Here two settings deserve a special mention: activating the third mouse button via ‘`Extensions`’ and the item Wheel Mouse. If you only have a two button mouse, then you should activate the item **`Emulate 3 buttons`**. Then you can simulate a third mouse button by pressing on both mouse buttons simultaneously.

If you are left-handed and would like to switch the order of the mouse buttons, you can do this later on. You don’t need to make any special settings for this here.

If you want to use the wheel on your “Intellimouse” set, in **`Wheel Mouse`**, the item **`Z Axis Mapping`** to **`Buttons`**. So that the X server “knows” what it should do when the wheel is moved, you must assign 'Negative movement' and 'Positive movement' to a mouse button. As buttons **1**, **2** and **3** are already used for the normal buttons, you should use buttons **4** and **5** for 'Positive' and 'Negative movement'. In the configured X Window System you can then, with the program imwheel, package imwheel, series ap (applications which do not need X11) pass on movements on the wheel to X11 applications. You should also read Section 5.6 page 162.

Close this dialog by selecting the item 'Next'.

Keyboard

As you might imagine, here you can set the keyboard language. If you have a "Win95" keyboard, this has 105 keys, and so you should choose **Standard 105 Key** in the menu 'Keyboard Model'. Under 'Keyboard Language' you should set the appropriate language. If you have a keyboard without "Windows", then you should choose the 101/102 key keyboard. Even if you have a laptop or a special type of keyboard with more or less keys than this, you can still choose the 102 or 105 key keyboard, since the key codes generated will match those of the standard PC keyboard in practically all cases.

If you click on 'Apply', the keyboard selected will be activated. In case you don't have an English keyboard, you should check to see if the keyboard layout is set with *dead keys*. You can check this by pressing the AltGr and + keys together. If the tilde sign does not appear immediately, but only after pressing the space bar, then the "dead keys" are activated. With dead keys you can easily add diacritical marks. Pressing AltGr and + and then n will produce "ñ" on the screen.

You can toggle the dead keys via the 'Expert' menu under the item 'Eliminate Dead Keys'. Set this item to **Yes** to deactivate the dead keys. After configuring the keyboard you can move to the next dialog on configuring search paths, by pressing the 'Next' button.

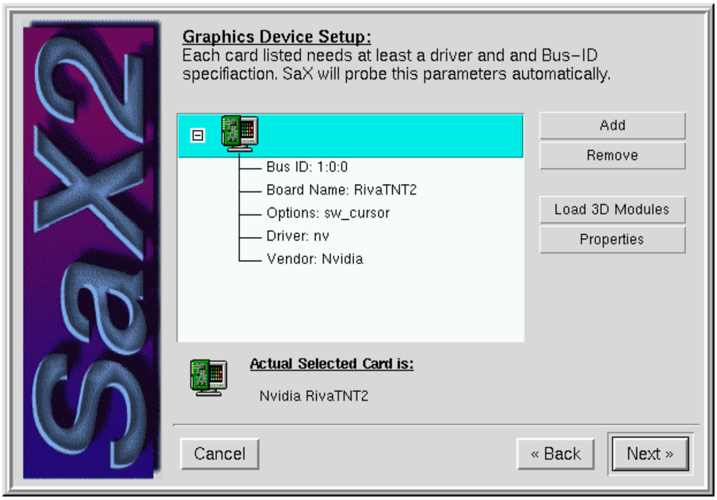
Search paths

You can now change the search path for fonts and change various "server flags". To move a font directory up or down, mark it with the left mouse button, hold the button down and move the line up or down, bringing this directory more to the front or the rear of the search path. With the 'New' and 'Delete' buttons you can add or remove directories accordingly.

Please note that the X server needs at least the directory with the "misc" character fonts, otherwise you will receive an error message from the X server when you try and start it. Section 5.6.1 page 166 describes how you can integrate further fonts into the X Window System and process texts in other languages.

Leave this dialog by clicking on the item 'Next'. The dialog for configuring the graphics card will appear.

Configuring the graphics card(s)



Graphics Device Setup:
Each card listed needs at least a driver and and Bus–ID specifiaction. SaX will probe this parameters automatically.

- Bus ID: 1:0:0
- Board Name: RivaTNT2
- Options: sw_cursor
- Driver: nv
- Vendor: Nvidia

Actual Selected Card is:
Nvidia RivaTNT2

Figure 5.4: SaX2: Configuring the graphics card(s)

SaX2 carries out a scan of the PCI and AGP bus and displays the graphics cards found in the field '`Card Tree`'. As a rule, you will see the graphics card which is installed in your machine. Via a database contained in SaX2, the matching XFree86-4.0 driver module is selected and displayed in the field '`Card properties`' under **`Driver`**. Under '`Videoram`' you can define the video memory size of the graphics card.

One innovation in XFree86-4.0 is the possibility of permanently allocating a driver to a card in a specific slot. SaX2 can carry out this allocation: normally this simply needs to be applied. For your information the slot is displayed in '`Bus ID`'. If, however, you later move the PCI graphics card to a different slot, then you will have to configure it again, since the **`Bus ID`** has been changed. To do this you can also set this item to `Single`, but you shouldn't use this setting if you have more than one graphics card in your computer.

Tip

ISA graphics cards cannot be detected automatically. They must be configured manually via '`Select Model`' or by directly selecting a driver. Only one ISA graphics card may be used in the same computer.

In '`Select Model`' you can manually select the graphics card, if it was not automatically detected. To do this click on '`Select model`' and choose the vendor and type of graphics card from the list.

Tip

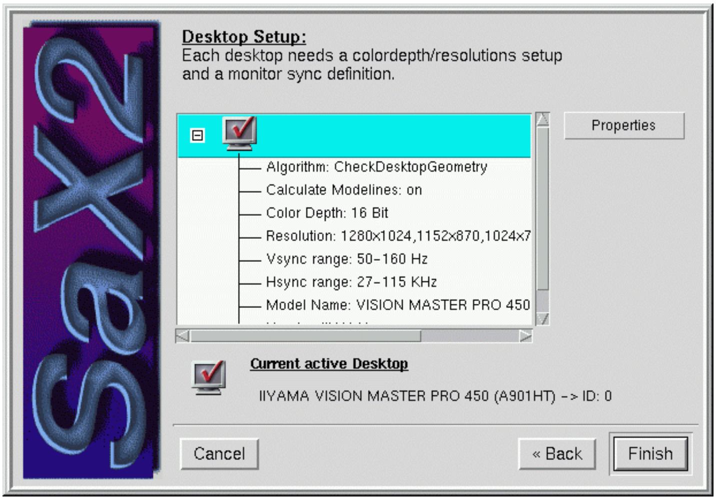
If your PCI or AGP graphics card is not supported by XFree86, you still almost always have the option of using the "Framebuffer driver".

If you have more than one graphics card installed, you can configure the other cards by clicking on the next graphics card symbol in the selection list. If this involves an ISA card, which cannot be automatically detected, you can add another graphics card by clicking on the 'New' button and configure this like the primary graphics card, by clicking on the symbol in the selection list.

In 'Expert' you will find further settings. Options such as **sw_cursor** which influence the behavior of the graphics driver can be set here under 'Option'. You normally only need special options if you can see presentation errors on the screen. For this please refer to Section 5.3.3 page 140.

Configuring the desktop

If you conclude the graphics card configuration by clicking on 'Next', you are given the chance to configure the screen. In this dialog you can adjust the monitor used and its resolution and color depth.



Desktop Setup:
Each desktop needs a colordepth/resolutions setup
and a monitor sync definition.

- Algorithm: CheckDesktopGeometry
- Calculate Modelines: on
- Color Depth: 16 Bit
- Resolution: 1280x1024,1152x870,1024x7
- Vsync range: 50-160 Hz
- Hsync range: 27-115 KHz
- Model Name: VISION MASTER PRO 450

Current active Desktop

IIYAMA VISION MASTER PRO 450 (A901HT) -> ID: 0

Figure 5.5: SaX2: Configuring the monitor and the desktop

For most graphics cards XFree86-4.0 can itself read DDC data from the monitor which it then displays in the selection dialog. You can see if automatic detection has worked by clicking on the small "+" symbol next to the monitor display. If the monitor was successfully "DDC probed", **DDC-probed** will be shown in the list alongside **Vendor** and **Name**. If the automatically set monitor frequencies do not work, they can be set manually. To do this select the item 'Select Model'.

A selection list appears with vendor names through which you can browse. If you select a vendor, the model names which can be chosen appear on the right-hand side. Highlight your monitor here and leave the dialog by pressing the'OK' button.

If you have a modern flat screen with LC display, or if you have or want to use the screen installed in your laptop, you should choose the item **LCD** **l***iquid* **c***rystal*

d*isplay* right at the top of the selection list. Through this the vertical frequency is limited to 60 Hz, which is usually the ideal refresh rate for LC displays.

If your monitor is not listed and the automatic detection went wrong, you can still enter your monitor data by hand. This is what the two fields are for in the section ‘`Monitor properties`’. Here you can set the allowed limits of the vertical and horizontal frequencies. Normally this data will be listed in the “technical data” section of the monitor users manual.

Caution

You should take special care when manually entering the frequency limits. Above all, a too high horizontal frequency on some monitors can quickly cause damage to your monitor.

As a rule the screen resolution and color depth is already suggested in **`Color depth/Resolution`**. If you are using a TFT or flat screen with LC-display, then you should take care to select the screen resolution matching the display resolution, otherwise the image will have to be scaled. SaX2 is able to detect the resolution of a digitally connected TFT or DSTN display, and will generally tell you the display’s resolution.

In setting the color depth, the settings 16, 256, 32768, 65536 and 16.7 million colors at 24 and 32 bit are available. To get a decent picture, you should set at least 256 colors. If however, you are using the“generic” VGA driver (XFree86 driver name **`vga`**, sometimes also referred to as the **`VGA16`** server), then you can unfortunately only use 16 colors.

With a color depth of 24 bit with 16.7 million colors you have a choice between 24 and 32 bpp *bits per pixel*. These two items only differ in the way that internal graphics memory is used by the graphics card. The 4 bytes per pixel (32 bpp, “padded pixel mode”) is more easily managed than the 3 bytes per pixel (24 bpp, “packed pixel mode”). This means that graphic representation in 32 bpp mode is usually somewhat faster than in the 24 bpp mode of the graphics card. Some graphics cards cannot manage this “packed pixel mode” with 24 bpp. The 65535 colors setting usually represents a fairly good compromise between use of video memory and display quality.

With the ‘`Expert`’ index tab you can make further settings. If you happen to have an unusual resolution, you can specify it here by hand. You can also stop SaX2 from calculating the screen parameters itself (this item is called **`Calculate Modelines`**). If you deactivate this, the VESA modes programmed into the X server are used which, however, are limited to a maximum of 75 Hz picture refresh rate.

Tip

If an existing configuration was loaded when SaX2 was started, the modelines of this configuration are taken over and no new modeline parameters are calculated. If you are adding a new mode to an existing configuration and don’t want to use a VESA mode, then you must explicitly activate the item **`Calculate Modelines`**.

If you are using just one graphics card, clicking on ‘Next’ will take you directly to the X server test. Otherwise you can specify the server layout in the next step.

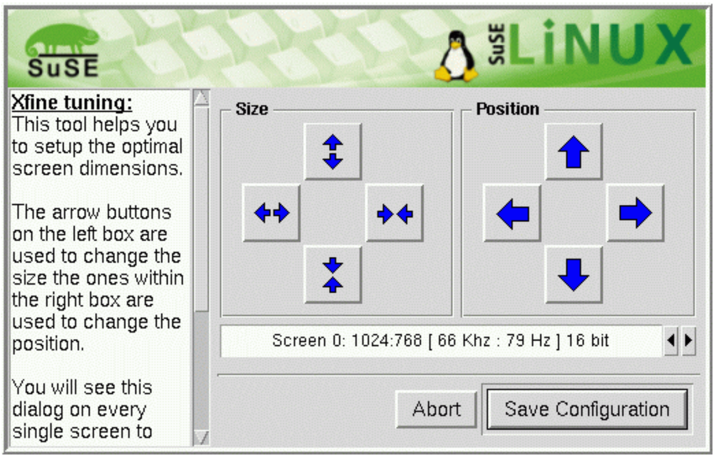
Layout

If you have more than one graphics card installed you will have the chance, after configuring the screen, to determine the screen distribution. You can specify in this dialog which monitor represents which “screen” and in what order these monitors are to be used (on top of or next to each other). The layout and the X server are adjusted accordingly. The ‘One Screen/Xinerama’ setting allows you to combine the two screens into one large desktop. This setting functions only if you are using the same resolution and color depth for both screens.

> **Tip**
>
> If you are using a “Multihead Setup”, that is, a configuration with 2 or more screens, then you should deactivate **Xinerama** the first time it is run, otherwise in the next step you will just get a fine-tuning dialog, which only allows you to configure the first graphics card. You should therefore switch off Xinerama in the first run through and only activate it later on.

Testing the configuration

Click — when you are ready — on the ‘OK’ button of the information box. After a short time the window shown here will appear.



Xfine tuning:
This tool helps you to setup the optimal screen dimensions.

The arrow buttons on the left box are used to change the size the ones within the right box are used to change the position.

You will see this dialog on every single screen to

Screen 0: 1024:768 [66 Khz : 79 Hz] 16 bit

Figure 5.6: SaX2: Testing the configuration

On the right-hand side you will find the two key fields, **Size** and **Position**, allowing you to adjust the picture. The arrow keys in the **Size** field allow you to stretch or squeeze the picture horizontally and vertically. In the **`Position`** field you can change the relative position of the picture on the screen. Use this to adjust the picture to your own requirements.

As a control that the picture is correctly adjusted, you will see four squares in the corner of the monitor screen. For an optimal picture position these should be completely visible without any false colors or distortion.

> Note
>
> If the test picture seems to be very dark, if you see a bright line on the edge of the picture, or if the picture overall is much too small, you should immediately press the keys Alt + Ctrl + ⟵. This will take you back to the SaX2 desktop configuration dialog. Then you must reduce the allowed frequencies of the monitor.

After correcting the picture position via the SaX2 settings or via the correction options provided by the monitor, you can end the test dialog.

To do this, select:

‘`Save`’: You will end your X Window Systemconfiguration and save the current settings. You return to SaX2. It will then ask you if you want to leave the program. The X Window Systemconfiguration is saved in the file `/etc/X11/XF86Config`. At the same time the X server used is set. This means that a link is created, `/var/X11R6/bin/X`, which points to the executable program, `/usr/X11R6/bin/XFree86`.

‘`Abort`’: This takes you back to the configuration dialogs of SaX2 without your settings being saved.

5.3.2 Reconfiguring

SaX2 also provides firm suppport if you want to customize a configured X server to your own requirements.

For this you can start SaX2 while the X server is running. If you wish, SaX2 can read the existing configuration file, `/etc/X11/XF86Config`, in which the X Window System saves and analyzes the configuration data. Then it is no longer necessary to go through the entire settings, from the mouse to the monitor, as SaX2 takes over the existing, functioning configuration of the X server. SaX2 then makes this data available in the above described menus. Note that you must explicitly select the reading in of the existing configuration.

However the user has all the freedom he wants in reconfiguring his X server. SaX2 also provides a convenient interface to manage the amount of color depth and resolutions of the graphics card, which can be quickly found via the dialog, ‘`Desktop setup`’.

5.3.3 Troubleshooting

This section is intended to help you configure the X server if there are problems getting the graphical interface to run at all, or if it runs but with a poor quality display.

Here the main and most frequent problems are discussed which can occur when configuring X with SaX2:

The graphics card is not supported

Using the framebuffer

Unfortunately there are not yet Linux drivers for all graphics cards. If your card is not supported, you can still usually get a graphical display using the *framebuffer* method. If your graphics card is VESA 2 compatible, this method should certainly work.

The advantage of this is that it works with almost every modern graphics card and also practically with every laptop. Because the graphics controller of the card is bypassed, the display will be unaccelerated. The X server simply accesses the *framebuffer* directly after the graphics mode is switched on when the kernel is started.

This works in the following way: when it starts the Linux kernel calls up — even before it switches to 32-bit protected mode — the VGA BIOS of the graphics card and instructs this to switch to a specific VESA graphics mode. The text depiction (that is the text console) then continues in this set graphics mode. The VGA BIOS is written in 16 bit code and therefore cannot be called up while the Linux system is running. As a result, this video mode specified at the start remains until the Linux system is shut down.

In order to now use this VESA framebuffer, support must exist in the kernel for it and the graphics mode must be selected when booting. The SuSE Linux kernel, of course, contains support for the VESA framebuffer. However you need to select the required graphics mode when booting the system. To do this enter the parameter **vga=x** at the LILO boot prompt, where **x** stands for a value to be taken from table 5.2.

Desired color depth	Resolution in pixels			
	640x480	800x600	1024x768	1280x1024
256 (8 bit)	769	771	773	775
32768 (15bit)	784	787	790	793
65536 (16bit)	785	788	791	794
16.7 Mill. (24bit)	786	789	792	795

Table 5.2: Possible VESA modes

You can also specify this parameter directly as a **vga** parameter in the file `/etc/lilo.conf`. The line **vga=x** should be included as a separate line in this configuration file, not as part of the **append** line. The installation of LILO is described in more detail in Section 4.4 page 98.

After starting the Linux system you should log in to the system as the user **root** and start SaX2 in the following way:

```
earth:/root # sax2 -m 0=fbdev
```

Note that the **0** here is a “zero”, and not a capital “o”.

This instructs the X server to use the driver for the framebuffer. Since resolution, color depth and the frequency rate is fixed, you can — if you are satisfied with mouse and keyboard settings — have SaX2 directly save the automatically detected data and leave SaX2 again.

Using the VGA16 server

If your graphics card is not capable of any of the VESA modes, or if you have an ISA card, you should still be able to configure it to run with the **vga** module. This will be suggested to you if the graphics driver is not automatically detected. Select the **vga** driver in SaX2 in ‘`Configure graphics card`’ under ‘`Driver`’. This will limit you in ‘`Desktop`’, to a resolution of 640x480 pixels with 16 colors (4 bit color depth). If your graphics card supports the VESA framebuffer, the framebuffer method described above is usually preferable, because of this limitation of the **vga** driver. To use the standard **vga** driver straight away when starting SaX2 enter the following command:

```
earth:/root # sax2 -m 0=vga
```

As this parameter only defines which X server SaX2 is to use during configuration, it can be useful to start the X server with this option with a graphics card which is supposed to be supported by XFree86.

In the case of certain “exotic” graphics cards, SaX2 may fail to start in the correct mode initially: if so, do the configuration using the vga module, but then choose the necessary options for the required driver module in the ‘`Expert`’ menu of the ‘`Graphics cards`’ dialog.

Selecting the mouse directly when starting SaX2

If SaX2 is unable to detect your mouse automatically, you can specify the mouse type when starting SaX2, thus avoiding the automatic mouse detection. For this, there are two command line parameters to determine the mouse protocol and port used. The **-t** parameter is used for defining the protocol used, and the **-n** parameter for the device file of the mouse. The following example configures an Intellimouse connected to the PS/2 port:

```
root@earth:/root > sax2 -t imps/2 -n /dev/psaux
```

Refer to the available protocols shown in table 5.1 page 132 in Section 5.3.1 page 131 and the manpage on the configuration file of the X server, **XF86Config**.

Correcting various display errors

In general, SaX2 will configure the graphics card correctly: however there are occasional cases where the display quality is poor. This can be corrected by changing an option in the X server configuration file `/etc/X11/XF86Config`.

This section aims to illustrate errors which occasionally occur when the X server is running and also to provide possible solutions. As a rule the graphic depiction is error-free after configuring with SaX2, but when using certain graphics cards depiction errors may occur which can be removed using an option in the X server configuration file, `/etc/X11/XF86Config`.

Sometimes depiction errors are also caused by hardware limitations; especially if the graphics card uses cheap DRAM memory, it is possible that the bandwidth of the memory is simply no longer adequate for high resolutions with 32 bpp and high frequencies.

Sometimes optimized BIOS settings will get their "revenge": Linux or the X server does not "like" the BIOS setting **`Video memory cache mode`** which other new BIOS versions provide. Everything functions normally in text mode, but apparently the chip set of the computer then "optimizes" access to the input/output register of the graphics card which can completely mess up the picture with the wrong colors when the X server starts and sometimes cause a total system crash.

If the X server does not start at all, this is almost always because of a syntax or logical error in the configuration file `/etc/X11/XF86Config` and it is possible that the hardware specified in the configuration file cannot be found in the computer. It can be useful to look at the X server logfile and refer also to Section 5.3.4 page 143 when doing this.

Solutions to various possible problems are outlined below.

- Many display problems are caused by the vertical synchronization frequency (`vsync`) being set too high, which results in incorrect values for `hsync` and DotClock.

  A reduction of the picture frequency rate to 80 Hz, for instance, will still produce a picture which doesn't flutter, and which is also a great deal sharper than one set at 160 Hz. At a picture frequency rate of 80 Hz the pixel rate is only half of that at 160 Hz.

  You should, therefore, try reducing the maximum vertical frequency rate used. To do this, choose the '`Monitor`' tab in SaX and, in the '`Advanced`' section, set a lower maximum frequency rate. For modern monitors a normal picture frequency rate lies between 80 and 90 Hz.

- Sometimes there are also problems with the "hardware cursor". You will see a square block or something which looks like a "barcode" where the mouse should be. The solution to this: add the `Option "sw_cursor"` to the `Section "Device"`.

- The best place to look for information and hints on configuration is in the README files which are sorted by chipset manufacturers' names in the directory `/usr/X11R6/lib/X11/doc/`. The manpages of the X servers and the XFree86 FAQ are also useful (`http://www.xfree86.org`). The SuSE website also has information at:

  (`http://www.xfree86.org`) and the SuSE pages for updates:

  `http://www.suse.de/en/support/xsuse/`

The following table is ordered in terms of probability, i.e. as a rule you will be successful with step 1, otherwise with the second step. Step 3 is usually a last resort . . .

All the options are listed in the device section of the file `/etc/X11/XF86Config`.

- **No mouse cursor, but a colored, square "barcode".**
  1. Add the option `sw_cursor` to the `Section "Device"`.
  2. Add the option `no_imageblt` or `no_bitblt`.
  3. Add the option `noaccel`.
- **Picture is too narrow or squashed. The monitor settings are already at their limits.**
  1. The frequency rate or the `hsync` frequency is probably set too high and is at the limit of the monitor. Remedy: reduce the `vsync` and/or the `hsync` frequency.
  2. Adjust the picture with the program xvidtune. It is possible that just one mode is not quite correct.
  3. Add the parameters `+hsync +vsync` to the modeline and try replacing `+` with `-`.
- **When moving windows, stripes, "lumps" or parts of windows stay where they are. They do not disappear even when the movement is stopped. They only disappear when the desktop is refreshed.**
  1. Reduce the picture frequency rate or the resolution.
  2. Depending on the chip set, use the relevant options from the `README` files in `/usr/X11R6/lib/X11/doc/`. For example the options `fifo_conservative` or `slow_dram`. *Note:* the options required depend on the particular chipset.
  3. The `noaccel` option, but possibly the options `no_imageblt` or `no_bitblt` may be sufficient.
- **"Noise" – Image interference when moving windows or viewing videos which disappears when the image is stationary.**
  1. Reduce the frequency rate, color depth or resolution.
  2. Reduce the frequency rate of the card, or add or remove wait states. This sometimes works with `set_mclk` (not with all chip sets!). More detailed information can be found in the `README` directory. *Attention:* this option is dangerous, the card could be set at too high a frequency).
  3. It is possible that the bus rate is set too highly. Check the bus rate on the PCI/VLB or ISA bus.
- **When starting XFree86, the screen turns black.**
  1. Reduce the frequency rate.

2. Check the BIOS settings of the computer. Deactivate all “Optimization settings” of the BIOS. Consult your mainboard manual if in doubt. Common culprits are the options **`Video memory cache mode`**, **`AGP Aperture size`** and all options which control PCI bus access, such as **`PCI Peer concurrency`** or similar. You will nearly always find these settings in a menu called **`Advanced Chipset Features`**.
3. Consider other possible sources of error: check the system for IRQ conflicts (e. g. PS/2 mouse needing IRQ 12).

5.3.4 The X server logfile

To analyze problems with the X server there is a very detailed log file which the X server creates when it starts. This file is created by the XFree86-4.0 X server according to the following pattern `/var/log/XFree86.Display.Screennummer.log`. If you start just one X server (which is the normal case) and this display is assigned the number “0”, the filename of this log file will usually be called `/var/log/XFree86.0.log`. Note that SaX2 here is an exception to this rule: here, at least temporarily, *two* X servers are running (Display **`:0`**) for the configuration dialogs and later a second one (Display **`:1`**) to test settings.

You should be aware that the format of this file has changed drastically compared to XFree86-3.3.x. Now a much clearer distinction is made between information messages, values taken from the configuration file, data originating from the computer hardware, and warnings and errors.

SaX2 allows you to view the X server log file with the middle mouse button. Here the various types of messages (error, informal, warning, etc.) are shown in different colors. This helps you to find the problem quickly if the X server does not start.

In general the beginning of such a log file appears as shown in 5.3.1.

```
XFree86 Version 4.0 / X Window System
(protocol Version 11, revision 0, vendor release 6400)
Release Date: 8 March 2000
        If the server is older than 6-12 months, or if your card is
        newer than the above date, look for a newer version before
        reporting problems.  (see http://www.XFree86.Org/FAQ)
Operating System: Linux 2.2.13 i686 [ELF] SuSE
Module Loader present
(==) Log file: "/var/log/XFree86.0.log", Time: Sat May 20 13:42:15 2000
(==) Using config file: "/etc/X11/XF86Config"
Markers: (--) probed, (**) from config file, (==) default setting,
         (++) from command line, (!!) notice, (II) informational,
         (WW) warning, (EE) error, (??) unknown.
(==) ServerLayout "Layout[all]"
(**) |-->Screen "Screen[0]" (0)
(**) |   |-->Monitor "Monitor[0]"
```

File contents 5.3.1: Extract from the X server log file

This provides you with the following information:

This is an XFree86 X server in version 4.0 compatible with X11R6.4 "vendor release 6400". The release date is 8th March, 2000.
The line **`Operating System:  Linux 2.2.13 i686 [ELF] SuSE`** refers to the system on which the X server was compiled. The kernel version and CPU definition can thus be different from your own system.

After these version messages the first login entries appear which the X server creates when it starts. First of all, is this the correct log file? Next to **`Time:`** the time is specified when the log file was created. Sometimes you might be searching in the wrong log file `;-)`

The same thing is valid for the configuration file if you did not specify a different file at the command line, this will always be `/etc/X11/XF86Config` on a normal SuSE Linux system.

The following table 5.3 explains the meaning of the two bracket characters at the beginning of further lines:

Symbol	Meaning
`(==)`	Defaults of the X server
`(--)`	Values taken from the system by automatic hardware detection.
`(**)`	Settings fixed in the configuration file.
`(++)`	Parameters which you have entered at the command line.
`(!!)`	Here the X server tells you in detail what it "is doing".
`(II)`	Version numbers of X server modules, etc. are usually recorded as "informational messages".
`(WW)`	Warnings: here the X server tells you why it is not carrying out certain actions specified in the configuration file or which should be activated by default.
`(EE)`	Error! These messages lead to the start procedure or the X server crashing. Look out for lines in the log file starting with **`(EE)`** if the X server does not start. You can remedy most errors yourself by means of these messages.

Table 5.3: Message types in the X server log file

You can also check the log file within the SaX2 configuration program by pressing the middle mouse button. Errors and warnings of the X server are highlighted in color. This file is also displayed if the X server which is called up by SaX2 for test purposes does not start or breaks off with an error.

If something unexpected happens when starting SaX2 or during the configuration steps, then all errors and steps concerning SaX2 are logged in the file `/var/log/SaX.log`. X server errors are logged, as described above, in the file `/var/log/XFree86.0.log`. By means of these files you will find clues on how you can carry on from here.

5.3.5 Starting the X Window System

The X Window System is started with the command startx. A preconfigured GUI for KDE is provided for the sample user. We recommend you starting the X Window System from this account, and *not* as `‘root’`. X11 server error messages are saved in the `~/.X.err` file. The **`startx`** command has a few options; for instance, you can select 16 bit color depth by typing

```
newbie@earth: > startx -- -bpp 16
```

5.4 Configuration Using SaX

The original version of SaX (***S**uSE **A**dvanced **X** Configuration **T**ool*) – as opposed to SaX2 – is the configuration tool for the versions 3.3.x of XFree86. If you are using XFree86 version 4.0 you should use SaX2 as described above. If for some reason you need to use XFree86-3.3.x, you should follow the instructions below.

> Note
>
> You have to start SaX (**sax**) as user `'root'`. You can also start SaX from YaST : '`System administration`' and then '`Configure XFree86[tm]`' (cf. Section 3.6 page 78).

The program is started from the command line by typing:

```
earth:/root # sax
```

As soon as the program starts it looks for any PCI cards which are installed. If a PCI graphics card is found, it will be identified and displayed in the program under '`graphics card`'.

After the PCI scan, the main window is opened to present the tab windows for the mouse ('`mouse`'), keyboard ('`keyboard`'), graphics card ('`graphics card`'), monitor ('`monitor`') and desktop ('`desktop`'). (You can change to each window by simply clicking on the appropriate tab). SaX then loads its own hardware database (this may take a few seconds). The data found during the system scan will be presented in its respective categories; the relevant graphics card, for example, can be found in the tab window '`desktop`'.

SaX will do its best to recognize what hardware it can, but to be sure that the configuration settings are correct, you should check them all and modify them if necessary.

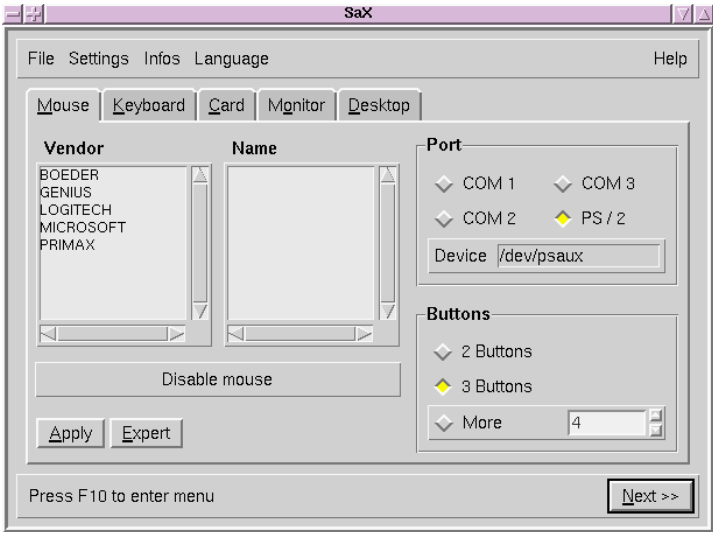
Essentially, the program provides you with five "index cards", i.e. '`Mouse`', '`Keyboard`', '`Graphics Card`', '`Monitor`' and '`Desktop`'. All cards can be accessed simply by clicking on the appropriate title ("tab" of the card in question).

If your mouse is not yet correctly configured, you can still access SaX via the keyboard. Pressing the `Tab` key repeatedly will cycle through each of the entry fields. To change to a different tab window, press `Tab` until the desired tab window title is highlighted with a black frame; then you can select the desired tab window using ← or →, and enter it by pressing ↵ (= `Enter`). In each tab window there are a number of selection elements, such as buttons, listboxes and entry fields. These can also be accessed by keyboard. To use a button (e. g. '`Apply`'), press `Tab` repeatedly until the desired button is highlighted. Pressing ↵ or ␣ activates the button; the desired action is carried out.

To select an entry in a listbox, push the `Tab` button until the desired box is highlighted. By pressing ↑ or ↓ you can look for an entry by way of its coloring, pressing `Enter` will then activate it.

The Mouse

The 'Mouse' tab window is the first to appear when the program starts (Figure 5.7).



BOEDER
GENIUS
LOGITECH
MICROSOFT
PRIMAX

◇ COM 1 ◇ COM 3
◇ COM 2 ◆ PS / 2

Device /dev/psaux

◇ 2 Buttons
◆ 3 Buttons
◇ More 4

Press F10 to enter menu

Figure 5.7: SaX: Mouse settings

If you have already configured your mouse when first installing Linux, e.g. while setting up gpm, these settings will be used by SaX and your mouse will work immediately in the X Window System – then you can ignore this.

If you have not configured your mouse, you should do it now. Press Tab twice and choose the relevant mouse type in 'Vendor' (move through the list with ↑ and ↓); use Enter to set the correct vendor name. Go to the selection list using Tab and choose the relevant type. If you press the 'Apply' button you can check to see if your choice was correct. The mouse cursor should move across the screen.

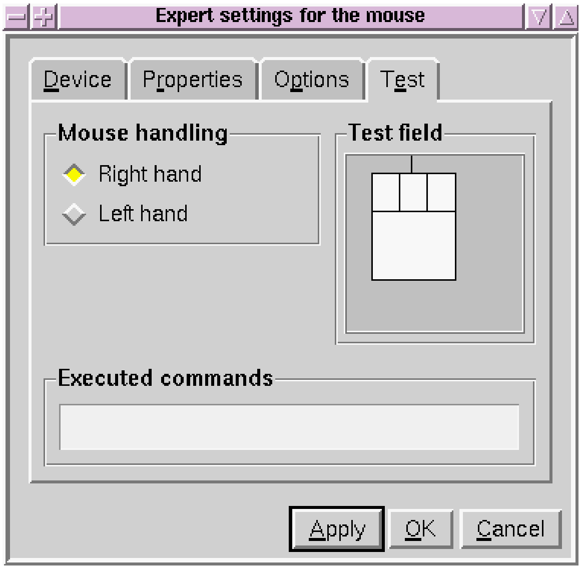
If you're not sure exactly what mouse type you are currently running on your system, your mouse is not included in the list, or if a serial mouse 'Microsoft' cannot run under the 'Standard Mouse' protocol, choose the sub-menu 'expert' to set the mouse protocol directly. There you can set further options, such as the baud rate and "three-button emulation".

Via 'Expert' the following tab windows are available:

'driver': If the vendor is not known, then the mouse protocol can be set here. The device file must also be selected. If you have a bus mouse you can try out the corresponding PS/2 variety.

'Properties': fine-tuning settings, determining how fast the mouse should react.

'options' 3-button emulation, etc.



Mouse handling

● Right hand

○ Left hand

Figure 5.8: SaX: Expert Mouse Settings

'Test': You can test the mouse configuration using the lower part of the frame 'Test field' (Figure 5.8.) If the mouse has been installed correctly, the mouse button symbols on the screen should react when you click on them.

The Keyboard

A Windows 95/98 keyboard with an English keyboard mapping has been set as the default. (Figure 5.9 on the next page). If you want to you use another keyboard, you must enter the correct settings, since the keyboard is one of the few hardware components not to be recognized independently by the hardware scan.

By comparing your keyboard with the 'Keyboard Image' shown on the screen you will be able to find the right model connected to your system. Don't forget to set the 'language' to English, if this is not already the case. Next to the delete tab there is a test field where you can check the settings made for the keyboard by entering various characters and seeing if they come out properly.

The switch 'Use "Dead" keys' serves with e. g. German keyboards to enable all signs on the keyboard buttons under X.

You probably do not need the settings in 'Expert' . . .

The changes are put into effect by pressing the 'Apply' button.

The Graphics Card

On the 'card' tab window, select the card vendors in the left hand list and the card version in the right hand list (Figure 5.10 page 150). SaX tries to recognize

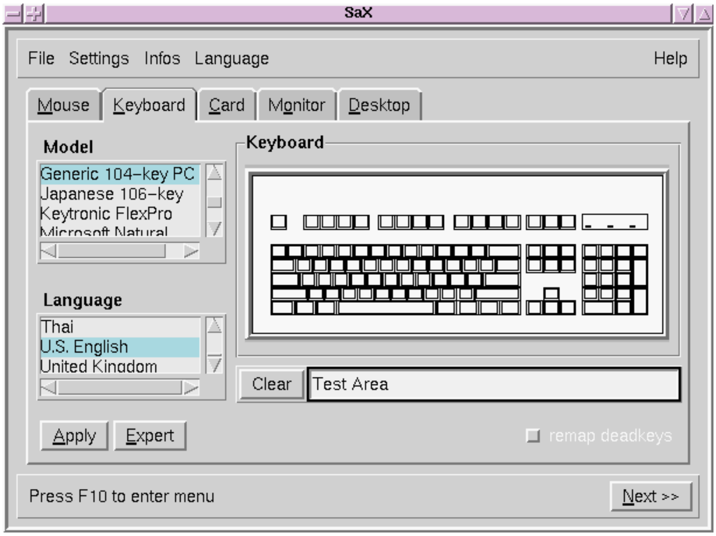


Figure 5.9: SaX: Keyboard

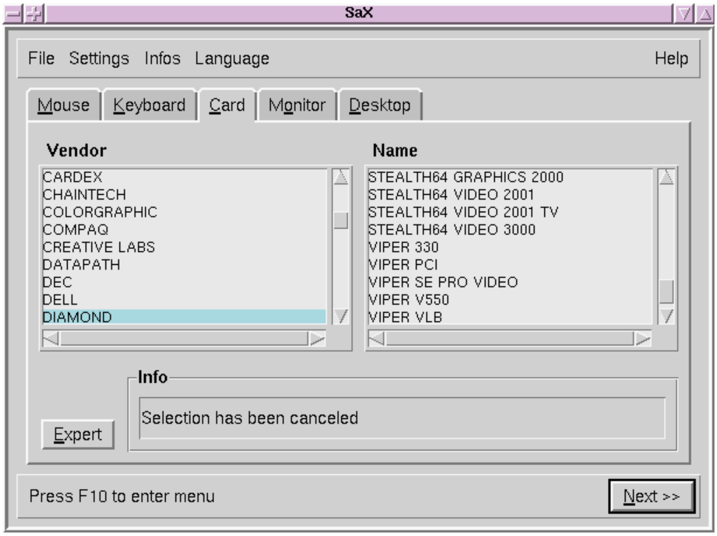
the graphics card independently, which nearly always works for PCI cards. The utility accesses an extensive database of current hardware to achieve this, which can also be found at `http://cdb.suse.de/` . Hardware that has been found will be highlighted in color.

> **Note**
>
> Sometimes a graphics card of the same series undergoes various "revisions", in which the hardware has been changed (different Ramdac). So it is possible that SaX may display values which are different to those in the manual of the graphics card. In these cases you should use the details from the hardware vendor's manual. Select the '`Expert`' button to do this.

There are advanced options hidden beneath the '`Expert`' button (Figure 5.11 page 151). These are relevant if you choose the X server directly ('`Server settings`'). If you want to set memory size or specify the RAMDAC value or if you would like to set a special RAMDAC or clock chip value (in '`chipsets`'), these options are also important. If the screen image looks distorted – if, for example, during a move-window operation, the window fragments or the window title suddenly blink – then you should reduce the RAMDAC value.

Some graphics cards need special '`options`' which can be found in this expert menu; normally they are not needed.

> **Tip**
>
> ISA cards will not be recognized "automagically"; for these, you have to select the relevant server "by hand".



Vendor:

CARDEX
CHAINTECH
COLORGRAPHIC
COMPAQ
CREATIVE LABS
DATAPATH
DEC
DELL
DIAMOND

Name:

STEALTH64 GRAPHICS 2000
STEALTH64 VIDEO 2001
STEALTH64 VIDEO 2001 TV
STEALTH64 VIDEO 3000
VIPER 330
VIPER PCI
VIPER SE PRO VIDEO
VIPER V550
VIPER VLB

Selection has been canceled

Figure 5.10: SaX: Graphics Card

If you see the error message `"The SVGA Server is not installed..."`, you must install the above-mentioned package via YaST (see Section 3.4 page 71).

The Monitor

The monitor settings are the last great hurdle on your road to a running X server. You find the same divisions in the left tab window list on '`Monitor`' to choose the monitor '`Vendor`'. Clicking one more option, you can choose your model on the right hand list ('`Type`'). Should you be unable to find your own monitor on the list, you can still enter horizontal and vertical frequencies specific to your monitor by pushing the '`Expert`' button. Normally you should be able to find these in your monitor handbook.

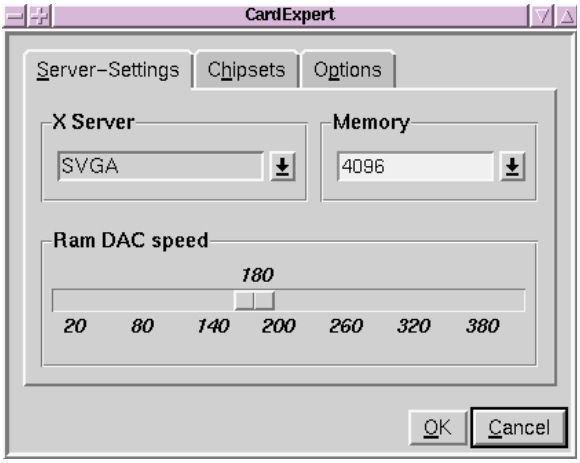
In case you do not have any monitor data available, SaX will set horizontal frequencies to 29-61 kHz and vertical frequencies will be set to 60-70 Hz. Most monitors should not be harmed by these settings.

If the screen remains dark for more than a few seconds after starting the X server or it flickers wildly, you should shut down the server immediately with `Ctrl` + `Alt` + `⟵` (`⟵` denotes the "backspace button") If you don't, this might damage to your monitor.

The Desktop

If your graphics card installation was successful, you will have a large number of resolutions and color depths to choose from. They can be administered from the '`desktop`' menu (Figure 5.12 page 152).

The '`desktop`' tab window might remind you of another operating system `;-)`



X Server: SVGA

Memory: 4096

Ram DAC speed: 180 (scale 20, 80, 140, 200, 260, 320, 380)

Figure 5.11: SaX: Graphics Card – Expert Options

In this window you can choose a different resolution (‘`resolutions`’) for each color depth (‘`color`’).

If you would like to set a list of several resolutions for a particular color depth, you can do this in expert mode (‘`Expert`’, Figure 5.13 page 153).

On the tab window ‘`Resolution`’ you will see:

‘`Resolution`’ This tab window is divided into three sections:

‘`Possible`’ resolutions.

‘`Current`’ resolutions.

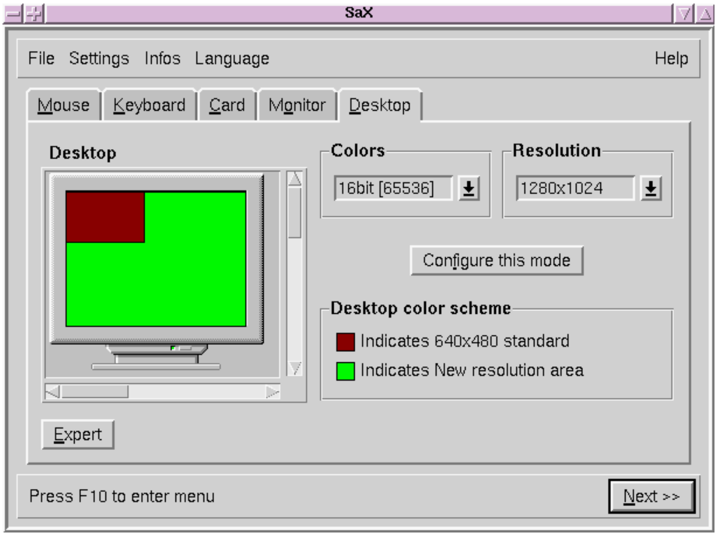
‘`Colors`’ the list of color depths.

First you should choose your preferred color depth on the vertical button panel to the right (‘`Colors`’); the resolutions have to be adjusted to these as well.

The X server can start in several color depths, e. g. in `8 bit` color depth. 256 colors can be shown on the desktop. Each color depth, in turn, can have different screen resolutions, e. g. `800x600`. Not all resolutions are available for each color depth. Thus for some cards a resolution of `1600x1200` pixels at `32-bit` color depth is not possible as too little graphics memory is available.

For `8 bit` color depth there are several resolutions available, from `1600x1200` down to `640x480`; they are listed in the ‘`Actual Resolution List`’. You can change resolutions by pressing `Ctrl` + `Alt` + `+` to cycle forwards through the list or `Ctrl` + `Alt` + `-` to go backwards. (you must use the keys `+` and `-` on the number pad). The first entry in the list is always chosen by the X server when it is initialized.

You must copy the desired resolution from the list of ‘`possible`’ resolutions to the list of ‘`current`’ resolutions. First you click on the resolution to mark it; then you click on the button ‘`I`’ to copy the desired resolution to the list of ‘`current`’ resolutions. By clicking ‘`R`’ you remove the current resolution.


SaX
File Settings Infos Language Help
Mouse Keyboard Card Monitor Desktop
Desktop
Configure this mode
Expert
Next >>

Colors: 16bit [65536]

Resolution: 1280x1024

Desktop color scheme

🟥 Indicates 640x480 standard

🟩 Indicates New resolution area

Press F10 to enter menu

Figure 5.12: SaX: Desktop

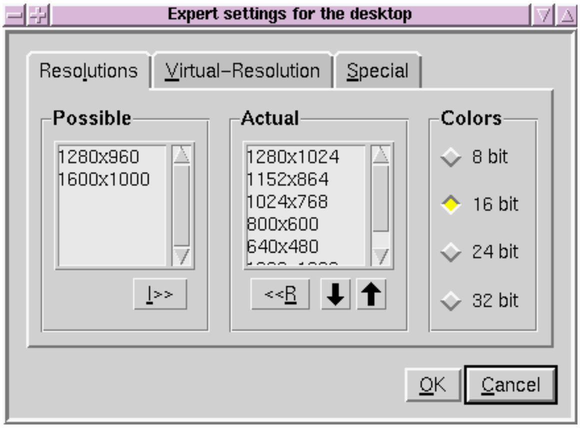
The sequence of '`current`' list entries can be changed via the arrow buttons at the lower end of the list window. First you mark the entry you want to move and then you left-click the up or down arrow. In this way the entry swaps places with its predecessor or successor.

When you are happy with your settings at `8 bit` color depth, you can select another color depth in the box on the right, such as `16 bit`. Now you will see a list of '`possible`' resolutions at this color depth. They can be changed in the same way as described above.

'`Virtual resolution`': this tab window lets you set the virtual resolution of your desktop. If you want to configure a virtual desktop you need to increase the values '`Virtual X`' (= width) and '`Virtual Y`' (= height). Generally, this feature is *not* chosen by most people. Background information: the X Window System is capable of defining a virtual desktop that is larger than the actual screen size, for example, using a virtual resolution of `1152x864` with a screen resolution of `800x600`.

'`Special`': the '`special`' tab window allows user-defined entries. In addition you can determine the '`quality`' of the "modelines", i.e. you can choose between two different calculation methods.

Now you should set the default color depth with which the X server should start. To do this you need to select the color depth you want in the '`screen`' tab window (with '`colors`') as your current color depth; once this is done you should move to the next item on '`Change current mode`'; this should start the X server check . . .



Possible
1280x960
1600x1000

Actual
1280x1024
1152x864
1024x768
800x600
640x480

Colors
○ 8 bit
● 16 bit
○ 24 bit
○ 32 bit

Figure 5.13: SaX: Screen

Testing the Configuration

After a short time a message box appears; if you are satisfied, click on '`OK`'. Then a background image and a split window should appear; in the left half there is information on your current resolution and on the horizontal and vertical monitor frequencies.

In the right half you should see two button fields, '`size`' and '`position`', which allow the image to be adjusted. The arrow boxes in '`size`' allow the horizontal and vertical image size to be increased or decreased; in the '`Position`' box you can change the position of the image relative to the monitor. Adjust the screen image to suit you.

Small rectangular image controls can be found in all four corners of the screen image. Ideally they should be all be visible and not display any distortion in color.

> **Tip**
>
> You can only make fairly small monitor adjustments using SaX; it cannot replace manual adjustment using the controls on the monitor itself !

After adjusting the screen image, there are two ways of closing the window:

'`Save`': End your X Window System configuration, saving the current settings. This takes you back to the command line.

'`Cancel`': Interrupt your X server setup and discard the settings.

Press `Alt` + `F1` if you want to return to the first console.

5.4.1 Reconfiguring

SaX can also be used to adjust an X server which is already running to your specific needs.

SaX reads, in the existing file `/etc/XF86Config`, where the X Window System stores and analyzes the configuration data. This is why you don't really need to make all the settings from the screen, since SaX adopts the existing, functioning configuration of the X server. SaX displays this data in the tab windows mentioned above.

You are free to do as you want, however, in making new settings for your X server configuration: you can select hardware from a comprehensive monitor database, thereby better adapting the configuration to the monitor's capabilities, as well as adjusting the screen position.

SaX also provides a comfortable GUI to administer color depths and resolutions. It is easily accessed, simply by clicking on the '`screen`' tab window.

5.4.2 Troubleshooting

Here are the main and most frequent problems encountered when configuring X windows with SaX:

- If the screen image flickers during the configuration test, or the image turns black, you must shut down the X server immediately, since the monitor could suffer damage if you continue to run the present configuration.

  Press `Ctrl` + `Alt` + ⟵

  You should select the '`Monitor`' tab window and look for a different monitor, or enter the monitor data by hand; the procedure is the same if the image begins to flicker while the screen is being adjusted.

- In particularly tricky cases, SaX has some command line options available, such as:

  `--server vga16`: This will start SaX with the VGA16 server instead of the server appropriate for your card. This server should run on almost all VGA cards. This server is automatically used if your graphics card is not recognized, or if you have an ISA card.

Up-to-date documentation on SaX can be found in the directory `/usr/share/doc/packages/sax`. If something unexpected happens when starting SaX or during the configuration steps, then this will be logged in the files `/root/ServerLog` and `/root/StartLog`. These files will provide you with clues on how best to proceed.

5.5 Configuration Using xf86config

In most cases, XF86Setup is superior to xf86config as a simple configuration tool. There are, nevertheless, some rare cases where XF86Setup fails. If so, just use xf86config, which almost always works.

XFree86-4.0 includes a similar text-based program, xf86config. At some points this contains dialogs which have been somewhat modified and, of course, it writes the configuration file to `/etc/X11/XF86Config`. In XFree86-4.0 however, the use of xf86config is usually not needed, since “problem” graphics cards can also be configured with the “Framebuffer” or with the **`vga`** module as described in Sections 5.3.3 page 139 and 5.3.3 page 140.

Make sure you have the following information available:

- mouse type, port to which the mouse is connected and baud rate (the baud rate is normally optional).
- specifications of the graphics card
- monitor data (frequencies, etc.)

If these settings are known, or you have your manuals at hand, you can start configuring. Remember that you have to be `‘root’` to do this.

The configuration is started with:

```
earth:/root # /usr/X11R6/bin/xf86config
```

Mouse

After the welcome screen, you are asked about your mouse type. You are offered the following selections (see Output 5.5.1:

```
 1. Microsoft compatible (2-button protocol)
 2. Mouse Systems (3-button protocol)
 3. Bus Mouse
 4. PS/2 Mouse
 5. Logitech Mouse (serial, old type, Logitech protocol)
 6. Logitech MouseMan (Microsoft compatible)
 7. MM Series
 8. MM HitTablet
```

Output 5.5.1: Mouse selection for X

While selecting the mouse, you should bear in mind that many of the new Logitech mice are either Microsoft compatible, or use the MouseMan protocol. The selection **`Bus Mouse`** refers to any bus mouse, including Logitech!

Selection is made by entering the relevant number. There may be a question whether “ChordMiddle” should be activated. This is necessary for some Logitech mice or trackballs, to activate the middle mouse button.

```
Please answer the following question with either ’y’ or ’n’.

Do you want to enable ChordMiddle?
```

If you have a two-button mouse, you can emulate the third button by answering 'y' to the next question.

```
Please answer the following question with either 'y' or 'n'.

Do you want to enable Emulate3Buttons?
```

The middle button is emulated by simultaneously pressing the two mouse buttons.

Next you must specify the mouse's interface:

```
Now give the full device name that the mouse is connected to, for example

/dev/tty00. Just pressing enter will use the default, /dev/mouse.

Mouse device:
```

If you have already entered a port for your mouse during the system installation, just enter **/dev/mouse**.

Keyboard

Next you are asked whether to assign **Meta** (ESC) to the left Alt-key and to assign **ModeShift** to the right Alt-key.

```
Please answer the following question with either 'y' or 'n'.

Do you want to enable these bindings for the Alt keys?
```

If you answer 'y', you can access the German keyboard via the right Alt, and the left Alt can serve as the meta key. [3]

Monitor

Next you should specify your monitor. You should be extremely careful with vertical and horizontal frequencies! These can be found in your monitor handbook.

> **Caution**
>
> Setting frequencies incorrectly can lead to irreparable damage to your monitor! The X Window System only addresses video modes which operate the monitor in the given frequency range. Entering frequencies for which the monitor was not designed can cause severe damage to it!

Some monitors are listed under /usr/X11R6/lib/X11/doc/Monitors [4].

To enter the horizontal frequency, the following selection is displayed (see Output 5.5.2 on the facing page):

You should only choose one of the predefined modes if you are unsure of the settings for your monitor. Selection '10' allows you to enter your own frequencies.

[3] e. g., in Emacs.

[4] We cannot be held liable, of course, if this information is inaccurate!

```
hsync in kHz; monitor type with characteristic modes
 1 31.5;               Standard VGA, 640x480 @ 60 Hz
 2 31.5 - 35.1;        Super VGA, 800x600 @ 56 Hz
 3 31.5, 35.5;         8514 Compatible, 1024x768 @ 87 Hz interl.
                       (no 800x600)
 4 31.5, 35.15, 35.5;  Super VGA, 1024x768 @ 87 Hz il.,
                       800x600 @ 56 Hz
 5 31.5 - 37.9;        Extended Super VGA, 800x600 @ 60 Hz,
                       640x480 @ 72 Hz
 6 31.5 - 48.5;        Non-Interlaced SVGA, 1024x768 @ 60 Hz,
                       800x600 @ 72 Hz
 7 31.5 - 57.0;        High Frequency SVGA, 1024x768 @ 70 Hz
 8 31.5 - 64.3;        Monitor that can do 1280x1024 @ 60 Hz
 9 31.5 - 79.0;        Monitor that can do 1280x1024 @ 74 Hz
10 Enter your own horizontal sync range
Enter your choice (1-10):
```

Output 5.5.2: Entry for the monitor’s horizontal frequency

The next screen asks you to enter your monitor’s vertical frequency (see Output 5.5.3). Again, using the known values (i.e., choice ‘`5`’) is preferable to using one of the items ‘`1`’ to ‘`4`’.

```
 1 50-70
 2 50-90
 3 50-100
 4 40-150
 5 Enter your own vertical sync range

 Enter your choice (1-5):
```

Output 5.5.3: Vertical frequency choices

Next you should enter a name, vendor name and model for your monitor:

```
Enter an identifier for your monitor definition:
Enter the vendor name of your monitor:
Enter the model name of your monitor:
```

These are just descriptive names used to document your configuration which do not affect the configuration itself. Merely pressing `Enter` will select the default values which are usually sufficient.

Your monitor configuration is now complete.

Graphics Cards / X server

Next you must specify your graphics card:

```
Do you want to look at the card database?
```

If you enter ‘`y`’, a selection of predefined cards is presented.

Here you can select your card by pressing the corresponding number. Do not trust this list blindly, since there can be differences in clock chip and RAMDAC[5] settings!

[5] **R**andom **A**ccess **M**emory **D**igital-to-**A**nalogue **C**onverter.

This is why there is a menu item later to select a RAMDAC and a clock chip, even though you have entered them already. Then the predefined settings for this card will be presented as an extra option.

The card definitions contain information on clock chips, RAMDAC and the X server to be used. Furthermore, some valuable information concerning the card is written to the device section in `XF86Config`.

If your card is not listed, do not panic. You should switch back to the normal configuration by selecting ‘`q`’. Only select one of the defined cards if it matches your card exactly! Selecting a card with a similar name is not recommended. Similar names do not necessarily refer to similar hardware.

Further information on how to configure your card is given in chapter Section 5.6 page 162.

After specifying your card, the X server is next. xf86config displays the choices, as seen in Output 5.5.4.

Choice ‘`5`’ only appears if you have selected one of the predefined cards in the previous step. In this case, choose ‘`5`’ to select the X server most suitable for your card.

```
1 The XF86_Mono server. This a monochrome server that should work on
  any VGA-compatible card, in 640x480 (more on some SVGA chipsets).
2 The XF86_VGA16 server. This is a 16-color VGA server that should
  work on any VGA-compatible card.
3 The XF86_SVGA server. This is a 256 color SVGA server that supports
  a number of SVGA chipsets. It is accelerated on some Cirrus and WD
  chipsets; it supports 16/32-bit color on certain Cirrus
  configurations.
4 The accelerated servers. These include XF86_S3, XF86_Mach32,
  XF86_Mach8, XF86_8514, XF86_P9000, XF86_AGX, XF86_W32 and
  XF86_Mach64.

These four server types correspond to the four different "Screen"
sections in XF86Config (vga2, vga16, svga, accel).

5 Choose the server from the card definition, XF86_S3.

Which one of these four screen types do you intend to run
by default (1-4)?
```

Output 5.5.4: Selecting an X server

When you have selected a server, you are asked if you want to create a symbolic link to `/usr/X11R6/bin/X`. If you answer with ‘`y`’, you are asked whether you want to put it in `/var/X11R6/bin/X`.

```
Do you want to set it in /var/X11R6/bin?
```

Reply with ‘`y`’, since it may not always be possible to write to `/usr`.

Afterwards, if you have selected ‘`4`’ (the accelerated servers) in the previous selection, a menu is presented of all available accelerated X servers, as shown in Output 5.5.5 on the facing page.

After selecting your X server, you now have to configure your graphics. First you should specify the amount of memory the card has, as seen in Output 5.5.6 on the next page.

```
Select an accel server:

1 XF86_S3
2 XF86_Mach32
3 XF86_Mach8
4 XF86_8514
5 XF86_P9000
6 XF86_AGX
7 XF86_W32
8 XF86_MACH64

Which accel server:
```

.

Output 5.5.5: Accelerated X server options

```
How much memory do you have on your graphics card:

1 256K
2 512K
3 1024K
4 2048K
5 4096K
6 Other

Enter your choice:
```

Output 5.5.6: Selecting video memory

Next, you must enter the name, vendor name and type for your graphics card. These are merely descriptive entries. If you earlier selected a card from the predefined list, pressing Enter will enter this as the default.

```
Enter an identifier for your graphics card definition:

Enter the vendor name of your graphics card:

Enter the model (board) name of your graphics card :
```

If you chose an accelerated X server, you must enter the RAMDAC settings. This only applies to the S3 and AGX servers.

In most cases, simply pressing Enter will suffice. If you have selected a graphics card that supports a specific RAMDAC, this should be chosen here (see Output 5.5.7 on the following page).

After answering this question, you can enter a clock chip for accelerated cards, if you have one (see Output 5.5.8 on the next page). Entering a clock chip avoids clock lines, as the clocks needed can be programmed.

If a card without a clock chip is selected, just press Enter (thus not selecting a clock chip). If a card has been selected, the clock chip is set as default (if there is one).

If no clock chip has been set, xf86config suggests running X -probeonly to determine the clock timings supported. These are automatically written in `XF86Config` in a separate *clocks* line.

Here, we must explain why the automatically defined settings can be *really dan-*

```
 1  AT&T 20C490 (S3 server)            att20c490
 2  AT&T 20C498/21C498/22C498 (S3)     att20c498
 3  AT&T 20C505 (S3)                   att20c505
 4  BrookTree BT481 (AGX)              bt481
 5  BrookTree BT482 (AGX)              bt482
 6  BrookTree BT485/9485 (S3)          bt485
 7  Sierra SC15025 (S3, AGX)           sc15025
 8  S3 GenDAC (86C708) (autodetected)  s3gendac
 9  S3 SDAC (86C716) (autodetected)    sdac
10  STG-1700 (S3)                      stg1700
11  TI 3020 (S3)                       ti3020
12  TI 3025 (S3)                       ti3025
```

Output 5.5.7: Setting a RAMDAC

```
 1  AT&T 20C490 (S3 server)            att20c490
 2  AT&T 20C498/21C498/22C498 (S3)     att20c498
 3  AT&T 20C505 (S3)                   att20c505
 4  BrookTree BT481 (AGX)              bt481
 5  BrookTree BT482 (AGX)              bt482
 6  BrookTree BT485/9485 (S3)          bt485
 7  Sierra SC15025 (S3, AGX)           sc15025
 8  S3 GenDAC (86C708) (autodetected)  s3gendac
 9  S3 SDAC (86C716) (autodetected)    s3_sdac
10  STG-1700 (S3)                      stg1700
11  TI 3020 (S3)                       ti3020
12  TI 3025 (S3)                       ti3025
13  Normal DAC                         normal
```

Output 5.5.8: Setting the clock chip

gerous: if the card has a programmable clock chip, the X server, when probing, cannot distinguish between the different clocks and only recognizes clocks 0, 1, and sometimes, 2. All other values are more or less random numbers (normally, clocks 0, 1, and 2 are repeated and are replaced by zeros).

All clocks apart from 0 and 1 are strongly influenced by the pre-programmed clock chip. Thus, clock 2 could have a different setting when probed (and which was written to the file `XF86Config`) than when the X server is later started. Then all the timings would be wrong and the monitor could be severely damaged!

A good indication of a programmable clock chip (and the problems this might entail) are many zeros or repeated timing values. Never ever write such values to `XF86Config`!

To configure clock chips, follow these steps:

- The best way is to enter an existing (*programmable*) clock chip if there is one. It will be programmed accordingly and your `XF86Config` will not contain clock lines. You can compare chips on the card with the chips offered in the menu. Most newer S3 cards have a programmable clock chip.

- If you *do not have a programmable* clock chip, you should launch `X` -probeonly and compare these values with those in the manual. If these values correspond (±2), you should enter them in `XF86Config`. If there are no hints in the manual, you can determine the values by running `X` -probeonly (this works best on an unloaded machine). Check whether the values are correct, since clock values cannot be determined for every card. (Many zeros or repeating values are a sign of invalid settings.) Enter the correct values into `XF86Config`. Do not omit any values; do not try to rearrange them or change them in any way. The values have to be entered in their exact order.

Exception: if the P9000 server is used, the order is irrelevant; just enter the modes for the desired clock in the *clocks line*.

- In general: if there is a programmable clock chip, there should be *no* clocks line in `XF86Config` (exception, P9000).
  For cards without a programmable clock chip, there should be a *clocks line* in `XF86Config`. This avoids the tedious (and sometimes even dangerous) testing at each startup. Furthermore, for cards with unreadable values, there are no invalid values and there is no risk to your monitor.

After having read the previous section, if you want to let clocks be recognized automatically, just answer ‘`y`’ to the following question:

```
Do you want me to run 'X -probeonly' now?
```

Now the screen will turn black and then a list of probed clocks will be presented, or a message will appear that no clocks could be found. If you have selected a clock chip, this question will not appear, since the clocks are then programmed automatically. In this case, this section is skipped.

> **Caution**
>
> If the previous question has been answered with ‘y’ and the screen remains black for more than 30 seconds, you should cancel testing immediately with Ctrl + Alt + ←, or Ctrl + c. If this does not work, switch off the monitor and the computer to prevent the hardware from being damaged!

Saving your Configuration

Now the configuration file has to be written. It is recommended that you write it to `/etc/XF86Config` to ensure that, even in a networking environment, each machine has its own configuration file—even if they share the `/usr` file system.

xf86config first suggests that it should write `XF86Config` to the current directory. You should answer ‘`no`’ to this:

```
Do you want it written to the current directory as 'XF86Config'?
```

Then you are asked where to save your configuration:

```
Please give a path + filename to write to:
```

Here, you should enter: “`/etc/XF86Config`”.

Now xf86config exits to the command line. This completes the configuration of the X Window System.

5.6 Optimizing the Installation of the X Window System

This section describes the configuration file, `/etc/X11/XF86Config`. Each *section* starts with the keyword **`Section <name of section>`** and ends with **`EndSection`**. Below there is a rough outline of the most important sections.

Afterwards you will learn how to integrate additional fonts, how you can configure input devices and how 3D acceleration is implemented. This is also managed in certain sections of the `XF86Config` file, of course, although integrating an additional font requires the help of external programs, which are included with SuSE Linux or are part of the default installation. The methods discussed here aim to illustrate the possibilities available and serve as an incentive, but they do not claim to cover all eventualities.

The programs SaX2 and xf86config (for XFree86-4.0) create the file `XF86Config`, by default in `/etc/X11`. This is the primary configuration file for the X Window System. You can find all the settings here concerning your graphics card, mouse and monitor.

`XF86Config` is divided into several sections, each one dealing with a certain aspect of the configuration. A section always has the same form:

```
Section <name of section>
  entry 1
  entry 2
  entry n
EndSection
```

The following types of sections exist:

`Files`	This section describes the paths used for fonts and the RGB color table.
`ServerFlags`	General switches are set here.
`InputDevice`	Input devices are configured in this section. In contrast to XFree86-3.3, both keyboards and mice as well as special input devices (touch pad, joysticks etc.) are configured via this section. Important terms here are **`Driver`** and the options which are defined by **`Protocol`** and **`Device`**.
`Monitor`	Describes the monitor used. The individual elements of this are the name, which is referred to later in the **`Screen`** definition, the **`bandwidth`** and the allowed sync frequencies (**`HorizSync`** and **`VertRefresh`**). Settings are given in MHz, kHz and Hz. Normally, the server refuses any modeline that does not correspond with the specification of the monitor. This is to prevent too high frequencies from being sent to the monitor by accident.

Table 5.4: continued overleaf...

`Modes`	The modeline parameters are stored here for the specific screen resolutions. These parameters can be calculated by SaX2 on the basis of the values given by the user and normally do not need to be changed. You can intervene manually at this point, however, if, for example you want to connect a fixed frequency monitor. An exact explanation of the individual parameters would be too much for this book. You can find details on the meaning of individual number values, however, in the HOWTO file `/usr/share/doc/howto/en/XFree86-Video-Timings-HOWTO.gz`.
`Device`	This section defines a specific graphics card. It is referenced by its descriptive name.
`Screen`	This section puts together a **`Driver`** (e. g., **`vga2`**), a **`monitor`** and a **`Device`** to form all the necessary settings for XFree86. In the **`Display`** subsection you can specify the size of the virtual screen (**`Virtual`**, the **`ViewPort`** and the **`Modes`**) used with this virtual screen.
`ServerLayout`	This section defines the layout of a single or multihead configuration. The input devices **`InputDevice`** and the display devices **`Screen`** are combined into one section.

Table 5.4: Sections in `/etc/X11/XF86Config`

We will now take a closer look at **`Monitor`**, **`Device`** and **`Screen`**. Further information on the other sections can be found in the manpage for **`XFree86`** (**`man XFree86`**) and manpage for **`XF86Config`** (**`man XF86Config`**).

There can be several different **`Monitor`** sections in `XF86Config`. Even multiple **`Screen`** sections are possible; which one is started depends on the server started.

Screen Section

First we will take a closer look at the screen section. As mentioned above, this combines a monitor with a device section and determines which resolution using which color depth, should be used.

A screen section might look like the example in File contents 5.6.1 on the following page.

The line **`Identifier`** (here **`Screen[0]`**) gives this section a defined name with which it can be uniquely referenced in the following **`ServerLayout`** section.

The lines **`Device`** and **`Monitor`**, specify the graphics card and the monitor which belong to this definition. These are just links to the Device and Monitor sections with their corresponding names or “identifiers”. These sections are discussed later in more detail.

Using **`DefaultColorDepth`**, you can select which color depth mode the server will start with if this is not explicitly stated.

```
Section "Screen"
    Driver      "accel"
    Device      "Miro Crystal 40SV"
    Monitor     "EIZO T563-T"
    DefaultColorDepth  16
    Subsection  "Display"
        Depth       8
        Modes       "1024x768" "800x600" "640x480"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
    Subsection "Display"
        Depth       16
        Modes       "1280x960" "1152x864" "1024x768" "800x600"
        ViewPort    0 0
        Virtual     1280 960
    EndSubsection
    Subsection "Display"
        Depth       32
        Modes       "1024x768" "800x600" "640x480"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
EndSection
```

File contents 5.6.1: An example of the screen section in `/etc/XF86Config`

There is a **`Display`** subsection for each color depth. **`Depth`** assigns the color depth which is valid for this subsection. Possible values for **`Depth`** are: 8, 16, 24 and 32. Not every X server supports all these modes. For most cards, 24 and 32 are basically the same, some take 24 for packed-pixel 24bpp mode, whereas others choose 32 for padded-pixel mode.

After the color depth, a list of resolutions is set (**`Modes`**). This list is checked by the server from left to right. For each resolution, a suitable **`Modeline`** is searched for, which has to correspond to one of the given clock rates or a clock rate to program the card.

The first resolution found is the so-called **`Default mode`**. With `Ctrl` + `Alt` + `gray +`, you can switch to the next resolution in the list to the right, with `Ctrl` + `Alt` + `gray -`, to the left, thus enabling you to vary the resolution while X Windows is running.

The last line of the **`Display`** subsection with **`Depth 16`** refers to the size of the virtual screen. The maximum possible size of a virtual screen depends on the amount of memory installed on the graphics card and the desired color depth, not on the maximum resolution of the monitor. Since modern graphics cards have a large amount of video memory you can create very large virtual desktops. You should note, however, that you may no longer be able to use 3D functionality if you fill practically the entire video memory with a virtual desktop. If the card has 16 MB video RAM, for example, the virtual screen can be up to 4096x4096 (!) pixels in size at 8-bit color depth. Especially for accelerated cards, however, it is not recommended to use up all your memory for the virtual screen since this memory on the card is also used for several font and graphics caches.

Device Section

A device section describes a specific graphics card. You can have as many device entries in XF86Config as you like, as long as their names are differentiated, using the keyword **Identifier**. As a rule — if you have more than one graphics card installed — the sections are simply numbered in order the first one is called **Device[0]**, the second one **Device[1]** etc. In the file 5.6.2 you can see the section from the **Device** section of a computer in which a Matrox Millenium PCI graphics card is installed.

```
Section "Device"
  BoardName     "MGA2064W"
  BusID         "0:19:0"
  Driver        "mga"
  Identifier    "Device[0]"
  VendorName    "Matrox"
  Option        "sw_cursor"
EndSection
```

File contents 5.6.2: The device section of the file /etc/X11/XF86Config

If you use SaX2 for configuring, then the device section should look something like the above diagram. Both the **Driver** and **BusID** of course are dependent on the hardware installed in your computer and are detected by SaX2 automatically. The **BusID** defines the PCI or AGP slot in which the graphics card is installed. This matches the ID which is displayed by the command lspci. Note here that the X server wants details in decimal form, while lspci displays these in hexadecimal form!

Via the **Driver** parameter you can specify the driver to be used for this graphics card. If the card is a Matrox Millenium then the driver module is called **mga**. The X server then searches through the **ModulePath** defined in the **Files** section in the drivers subdirectory. In a standard installation this is the directory /usr/X11R6/lib/modules/drivers. For this purpose simply _drv.o is added to the name, so in the case of the **mga** driver the driver file **mga_drv.o** is loaded.

The behavior of the X server or of the driver can also be influenced through additional options. An example of this is the option **sw_cursor** which is set in the device section. This deactivates the hardware mouse cursor and depicts the mouse cursor using software. Depending on the driver module, you have various options available which can be found in the description files on the driver modules in the directory /usr/X11R6/lib/X11/doc. Generally valid options can also be found in the manpage for **XF86Config** (**man XF86Config**) and manpage for **XFree86** (**man XFree86**).

Monitor Section

Monitor sections each describe, in the same way as the device sections, one monitor. The configuration file /etc/X11/XF86Config can again contain as many **Monitor** sections as you want in XF86Config. The server layout section specifies which monitor section is relevant.

The same applies for monitor definitions: they should only be set by experienced users. A critical part of the monitor section is the modelines which set horizontal and vertical timings for the appropriate resolution. The monitor properties, especially the allowed frequencies, are stored in the monitor section.

> **Caution**
>
> Unless you have an in-depth knowledge of monitor and graphics card functions, nothing should be changed in the modelines, since this could cause severe damage to your monitor!

For those who may want to develop their own monitor descriptions, the documentation in `/usr/X11/lib/X11/doc` might come in handy. The section [FCR93] deserves a special mention. It describes in detail how the hardware functions and how modelines are created.

Luckily a “manual” setting of the modelines is hardly ever needed nowadays. If you are using a modern multisync monitor, the allowed frequencies and optimal resolutions can, as a rule, be read directly from the the monitor by the X server via DDC, as described in the SaX2 configuration section. If this is not possible for some reason, you can also use one of the VESA modes included in the X server. This will function with practically all graphics card and monitor combinations.

5.6.1 Inserting Additional (True Type) Fonts

A standard X11R6 X server installation also includes a large number of fonts. These can be found in the directory `/usr/X11R6/lib/X11/fonts`, each divided into logically connected groups in subdirectories. Make sure that only subdirectories of the X server are used which –

- are entered in the files section, **`Files`** of the file `/etc/X11/XF86Config` as **`FontPath`**.
- contain a valid `fonts.dir` file.
- were not closed while the X server was running using the command xset −fp.
- or which were started while the X server was running using the command xset +fp.

Since version 4.0, XFree86 can use not only its own format **`Type1`** (a PostScript format) for scalable fonts and **`pcf`** for bitmap ones, but also the **`ttf`** (True Type font) of this file format. As described in Section 5.2 page 126, this support is provided via loadable modules of the X server. Thus you can also use directories containing True Type fonts together with the X server. To do this, hardly any preparation is needed.

A big advantage of most True Type fonts, apart from their very good scalability, is that these fonts almost always contain more than the normal 255 characters of the font for western Europe coded in “iso-8859-1”. With these fonts you can display Cyrillic, Greek or eastern European languages without any problem, and with special software, even Asian languages.

This description is essentially about the use of fonts as 8-bit character sets. If you want to use characters of Asian languages (Japanese, Chinese, etc.), you can use special editors which are also available in SuSE Linux.

An 8-bit character set contains 255 characters and basically consists of the US-ASCII character set, which defines only the first 128 of 255 possible characters, to be able to expand further characters. One text character thus occupies 8-bits in the computer memory. As 127 characters are certainly not enough to record the special characters, for example, of all European languages, the various languages are combined into groups and this group is then given a short name. The relevant character set is named according to the appropriate norm as the “iso-8859-x” character set, where by the x stands for a number from 1 to 15. The exact order of characters in the **`iso-8859-1`** character set can be found from the manpage for **`iso-8859-1`** (**`man iso-8859-1`**).

The more well-known codings are listed in table 5.5 : further ones can be taken from the above-mentioned manual page.

Font	Supported regions, contains special characters
iso-8859-1	West European languages: Spanish, German, French, Swedish, Finnish, Danish and others
iso-8859-2	Central and Eastern Europe: Czech, Rumanian, Polish, German and others
iso-8859-5	Cyrillic characters for Russian
iso-8859-7	Greek characters for Greek
iso-8859-15	As iso-8859-1, but with characters for Turkish and the Euro sign.

Table 5.5: Important font codings

The user must then — depending on the language used — select the matching coding. Especially when transferring texts between different computers, the coding used must also be transferred. The advantage of this procedure is obvious: To receive support for regional special characters you only need to select the correct coding and immediately (nearly) all programs will be able to portray these special characters, since (nearly) all programs use an 8-bit value (one byte) to represent a text character. If the wrong coding is chosen then the special characters will be wrongly depicted. With most X applications, as well as with the KDE desktop, you can usually select the coding of the character set when you are configuring the font to be used. In X applications the coding is usually referred to as **`Encoding`**.

The disadvantage of this method is that some language combinations are impossible: You can not, for example, easily write a German text with umlauts in which you mention Russian place names in Cyrillic.

This dilemma can only be solved using a different approach, with the use of Unicode. Unicode codes characters — differently to ASCII — not with one but with two or even more bytes, allowing considerably more characters to be represented. Only if you use Unicode can you depict Asian languages with more than 127 characters, such as Chinese, Japanese or Korean, on the computer. The disadvantage of this method is that most existing software cannot handle these characters and that you can only read or write texts yourself with Unicode characters using special software. If you want more information on using Unicode fonts in Linux, take a look at the URL http://www.unicode.org. It is expected that in future more and more programs will support Unicode characters. In SuSE Linux there is the program yudit to enter texts in Unicode. The program yudit can be found in package `yudit`, series `xap` or after installation via the SuSE menu, under **Office** and then under Editors.

After these observations, we now have a step by step description of the installation of additional fonts, using the example here of True Type fonts.

Locate the fonts that you want to install in your X Window System. If you have licensed True Type fonts then you can simply use these on your system. Mount the partition containing these fonts.

You should create a font directory — if this does not yet exist — and change to it. SuSE Linux already has a directory called `/usr/X11R6/lib/X11/fonts/truetype`, you can copy the relevant fonts to this directory.

```
earth:/root # cd /usr/X11R6/lib/X11/fonts/truetype
```

Create links to the ttf files and create the font directory. Note that for True Type fonts you need a special program called ttmkfdir, package `ttmkfdir`, series `xap` to create the file `fonts.dir`. Traditional X fonts are included using the command mkfontdir. Instead of the path `/path/to/the/fonts`, set the corresponding path in which these fonts are located.

```
earth:/usr/X11R6/lib/X11/fonts/truetype #
ln -s /pfad/zu/den/fonts/*.ttf .
earth:/usr/X11R6/lib/X11/fonts/truetype #
ttmkfdir -o fonts.dir
```

If the X server is already running, you can now make the fonts dynamically available. To do this enter:

```
newbie@earth:/home/newbie > xset +fp /usr/X11R6/lib/X11/fonts/truetype
```

Tip

The xset command accesses the X server via the X protocol. It must therefore have access permissions for the X server currently running. This is the case, for example, if the user `newbie`has started the X server. You can find more on this in the manpage for **xauth** (**man xauth**)

To set up the fonts permanently you should add this search path to the `XF86Config` file. You can use SaX2 to do this. To change the fonts path you must select the '`Custom`' configuration mode of SaX2. In '`Path dialog`' you can add the directory, with '`Add`', to the directories already listed.

You should test if the fonts were set up correctly. To do this use the command xlsfonts. If the fonts are correctly installed, the list of all installed fonts, including the newly installed True Type Fonts, is displayed.You can also use the KDE program, kfontmanager, which displays the installed fonts with an example text.

```
newbie@earth:/home/newbie > xlsfonts
```

These newly installed fonts can then be used in all X applications.

5.6.2 Setting up input devices

If you wanted to describe all the possible settings of input devices you could write a whole book on it, simply because there are a huge number of input devices available in the computer world.

The following section will try to give you a few recipes with which you can configure these input devices. The somewhat abstract term "input devices" refers to the mouse and keyboard, but also to touch pads, etc. Look for the relevant heading below to find more information on your relevant topic.

You can, of course, find more detailed information in the X server documentation, especially in the manpage for **XF86Config** (**man XF86Config**).

Mouse

Wheelmice If you have set up the mouse with SaX2, then you can also use the basic functionality. You might have a mouse with a wheel (a so-called "wheel mouse"), in which case you would like to use this wheel in X applications, e.g. for scrolling. If you have such a wheel mouse with SaX2 as described in Section 5.3.1 page 131, with the extended settings in the SaX2 mouse menu, 'wheel mouse', then movements on the wheel are passed on to applications as movements on buttons **4** and **5**. Unfortunately, only a few X applications can handle this. To be able to use the wheel in all programs, you just need to install the program package imwheel, series ap, in the running X server.

You can do this in the file ~/.xinitrc. You can insert the program command, for example directly under the commentary line finally start the window manager. This small utility converts the "mouse clicks" which are produced by moving the wheel, to configurable key clicks. These are preconfigured to PgUp and PgDn, the program can also be configured via the file /etc/imwheelrc. In this way you can operate every X program which can be used via the keyboard, with your wheel mouse as well. If you copy the configuration file to ~/.imwheelrc, then you can also make changes as a normal user, not just as 'root'.

Using the mouse for left-handed people To swap the left and right mouse button functions, you can enter the following command:

```
newbie@earth:/home/newbie > xmodmap -e "pointer = 3 2 1"
```

You can also include this command in the file ~/.xinitrc.

Two mice or a touch pad It is no problem if you want to include a second mouse or a touch pad. Again, we would advise you to use SaX2. Chose the ‘`Custom`’ configuration mode and in the ‘`mouse`’ menu, add a further mouse symbol, using the ‘`New`’ button. This second mouse will not be configured automatically, but you can enter the data yourself by hand. Under ‘`Drivers`’ you can chose the relevant driver **`mouse`** if this is another mouse, or you can select one of the drivers listed for touch pads. Under ‘`Protocol`’ and ‘`Connection`’, the procedure is similar. The setting ‘`Protocol`’ only makes sense if you are using the **`mouse`** driver. Otherwise you should select the pads type settings in the Expert dialog. If you have a touch pad driver, select **`None`** in the ‘`Protocol`’ section.

In the ‘`Expert`’ menu then choose additional options in ‘`Miscellaneous`’. The option **`Send Core Events`** deserves special mention here; if you activate this action you can use input devices in a parallel fashion. Otherwise you can switch between the two input devices using the command xsetpointer. You can use the command

```
newbie@earth:/home/newbie > xsetpointer -l
```

to have a list of possible input devices displayed.

Keyboard

By nature there is not as much to write about the subject of “Keyboard” as there is on other input devices. Usually there is nothing special to configure here.

The following programs and files can however be useful in certain cases: To make changes or just to test the keyboard layout during operation there is the program xkeycaps, package `xkeycaps`, series `xap` . In the file `~/.Xmodmap` you can then save changes permanently to individually changed characters. For the format of this file, see the manpage for **`xmodmap`** (**`man xmodmap`**).

In KDE the program kikbd (“International keyboard layout”) is available in the ‘`System`’ menu of KDE. With this you can switch very conveniently between different keyboard layouts. This program is very useful if, for example, you want to use an English keyboard for programming (here the brackets {} and [] are much more easily accessible) and a German keyboard to write texts.

5.6.3 3D acceleration

With a number of graphics cards it is now possible to use 3D acceleration with XFree86. Please note that a number of 3D drivers are still at a beta stage!

Unfortunately the newly introduced “DRI” *Direct rendering infrastructure* in XFree86-4.0 can only be used with the “Hacker kernel versions” 2.3. This **`DRI`** ensures that the X server can access the 3D acceleration functions of the graphics card directly with kernel support.

XFree86-3.3.x does not use any special kernel module for 3D support. Here there is a server module called **`glx.so`** which is configured in the **`Modules`** section of the file `/etc/XF86Config`. If you have not already activated 3D support during installation with YaST2, install the matching 3D module for your graphics card, using YaST. The module can be found in the series **x3d**.

This module provides support for 3D hardware acceleration. In the file /etc/XF86Config — if that was not already set up during installation — add the line **Load ”glx.so”** in the **”Modules” Section**. This module ensures that the X server understands the 3D commands and can interpret them correctly. A similar procedure is used in XFree86-4.0, only here you must omit the ending **.so**, as this ending is added automatically by the X server.

The 3D commands are then used by the OpenGL-compatible MESA graphics library. This library is in each case — always matching the graphics cards in question — contained in the same package as the **glx** module for the X server and accesses directly this 3D extension of the X server. This library therefore does not need to be installed again separately. Strictly speaking, the 3D support consists of 2 parts: a “shared library” which is installed in the system and which is used by applications like other shared libraries, and a modular extension of the X server.

Installed 3D applications can now directly use acceleration functions. A number of demo programs can be found in the package glutdemo, series x3d. After installing 3D acceleration, you can try for example, the following command in an xterm:

```
newbie@earth:/home/newbie > /usr/share/glutdemo/demos/atlantis/atlantis
```

6 Linux and Hardware

6.1 Preliminary Notes

It is now possible to integrate almost all hardware components into a Linux system with little (or more) trouble. How this is achieved in individual cases and which software is available is the subject of this chapter. For problems involving *printing*, see Chapter 8 page 205. *PCMCIA hardware* is described in Chapter 7 page 191.

6.2 Cards

In a standard PC, normally the PCI ("Peripheral Component Interconnect") bus is used. Most times the ISA ("Industry Standard Architecture") bus is still present in systems.

The AGP ("Accelerated Graphics Port") is an exception to this, as devices on this port are also visible on the PCI bus. They are treated logically in exactly the same way as PCI bus cards. PCMCIA bus cards are found mostly in portable computers such as laptops and notebooks. These cards are described in Chapter 7 page 191.

6.2.1 PCI and ISA Cards

We shall now take a closer look at PCI and ISA bus cards. The configuration of these cards will then be explained using sound cards as an example.

PCI cards

PCI bus cards normally don't need to be configured at all by the user. The agreement on the PCI bus standard was an attempt to remove many restrictions of the old bus system. In this respect, a sensible automatic configuration of the cards was planned. Each PCI card is activated by the computer's BIOS when booting. In many cases you can influence the distribution of the interrupts through settings in the computer BIOS. When Linux starts, it reads the configuration of the PCI devices directly from the PCI BIOS and, from this point on, uses this data for all information about the PCI subsystem.

With the command

```
earth:/ # lspci -tv
```

you can be shown a list of all the devices on the PCI bus recognized by Linux; see output 6.2.1.

```
-[00]-+-00.0  Intel Corporation 440BX/ZX - 82443BX/ZX Host bridge
      +-01.0-[01]----00.0  Nvidia Corporation Riva TNT
      +-04.0  Intel Corporation 82371AB PIIX4 ISA
      +-04.1  Intel Corporation 82371AB PIIX4 IDE
      +-04.2  Intel Corporation 82371AB PIIX4 USB
      +-04.3  Intel Corporation 82371AB PIIX4 ACPI
      +-06.0  Adaptec 7890
      09.0  Digital Equipment Corporation DECchip 21140 [FasterNet]
```

Output 6.2.1: Output of the command **`lspci -tv`**

In the following subsection we will also configure a PCI sound card.

ISA cards

The ISA card is the oldest form of PC-compatible card. The bus is 8 or 16 bits wide, and has a maximum frequency of 8 MHz. Many motherboards provide the option of increasing the frequency of the ISA bus – but this often leads to problems. ISA bus cards can be divided into three categories:

Legacy cards: These cards must be configured entirely by hand. Resources such as I/O addresses, interrupts and DMA channels must be set through jumpers or microswitches on the cards themselves. You must also ensure that resources such as IRQ or DMA channels are not used by two or even more cards simultaneously. Therefore the configuration of such cards often turns out to be somewhat difficult.

Jumperless cards: Jumperless cards, compared to the first generation of ISA cards, have the advantage that the setting of resources is no longer made via jumpers, but via a special configuration program. These programs, however, are usually only available for DOS, and thus cannot be used in Linux.

PnP cards: Plug and Play cards are the logical progression from jumperless cards. These cards contain data on their configuration and a list of possible configurations. A special program or a driver can now query the PnP cards in the system and configure all cards so they do not conflict with each other.

If a card has not been configured, then in practice it is not available to the system.

To configure PnP cards in Linux running a 2.2.x kernel, the package `isapnp` from the series `ap` is used. This package contains the two programs, pnpdump and isapnp.

How are PnP cards activated with isapnp tools?

Proceed as follows:

- Change to the user `'root'`.

- If the file `isapnp.conf` already exists in the directory `/etc`, you should first make a backup copy by changing its name.

  ```
  earth:/ # mv /etc/isapnp.conf /etc/isapnp.conf.bak
  ```

- By entering

  ```
  earth:/ # pnpdump -c > /etc/isapnp.conf,
  ```

  you will create the file `/etc/isapnp.conf`. The parameter **`-c`** ensures that pnpdump prepares the output file in such a way that the PnP devices are activated immediately.

- If you enter

  ```
  earth:/ # isapnp /etc/isapnp.conf,
  ```

  the PnP cards in your computer will be activated. – In SuSE Linux this is activated automatically each time the system is booted.

Possible sources of error

Symptom: **pnpdump** gives the output: `"No boards found"`. It may also be the case that one or more of the installed ISA PnP cards cannot be recognized.

Possible Explanations:

- There are no PnP cards installed on your system: check which cards are installed on your system and read the documentation about them. If necessary, ask the dealer from whom you bought your computer which cards are installed on it.
- The cards are defective: check if the cards are correctly positioned in the slots. See if the cards work correctly under a different operating system.
- There are certain cards which can either be run as ISA PnP cards, or which can be allocated fixed resources. (Example: a number of 10 MBit NE2000 ISA Ethernet cards). Usually you can switch between the two modes using a DOS program.

  Depending on the configuration of such cards, these do not appear in the output of **pnpdump**.
- There are a few ISA PnP cards which will not work together with other ISA PnP cards in the same system. In such cases, only one of the cards is visible in the output of pnpdump. It is possible that one of the cards can be configured so that it is no longer driven as an ISA PnP card (for example, with a jumper directly on the card so that it uses fixed resources).
- There seem to be (in very few cases) cards which perform a reset after pnpdump is run. If this involves a SCSI card, for example, a controller on which the root partition is dependent, this can cause the computer to hang. If this problem occurs, you must try and configure the card in such a way that it no longer functions as an ISA PnP card.

  Otherwise you can do away with initializing the card in Linux completely and boot the system with the help of loadlin; in this case, the card is already initialized.

Symptom: When running the command isapnp or when booting, you will see an error message like this:

```
* LD setting failed, this may not be a problem.
* Try adding (VERIFYLD N) to the top of your script
*
* Error occurred requested ‘LD2‘ on or around line 319
* --- further action aborted
```

Solution: Follow the advice of the error message, and at the beginning of the file `/etc/isapnp.conf`, insert the line

```
(VERIFYLD N)
```

that is:

```
# [...]
# (DEBUG)
(VERIFYLD N)
(READPORT 0x0203)
(ISOLATE)
(IDENTIFY *)
# [...]
```

Symptom: When running the command isapnp or when booting, you see an error message like this:

```
[...]
/etc/isapnp.conf:66 --
  Fatal - resource conflict allocating 16 bytes
  of IO at 220 (see /etc/isapnp.conf)
/etc/isapnp.conf:66 -- Fatal - IO range check
  attempted while device activated
/etc/isapnp.conf:66 -- Fatal - Error occurred
  executing request ’<IORESCHECK> ’ --- further
  action aborted
```

Solution: You probably have a conflict between the values selected in the file `/etc/isapnp.conf` and those resources already being used in your system. You should compare your `/etc/isapnp.conf` with the information on the resources in question, which can be found in `/proc`; this should not be necessary if you allow pnpdump to undertake this allocation itself by using the option **`-c`**.

Caution

In many cases difficulties can arise if, in the setup of the computer, the automatic configuration of ISA PnP resources is switched on through the BIOS. You should switch this off. Since the configuration is performed by means of pnpdump or isapnp, this setup option is not necessary.

In case of problems which go deeper than this, you should consult the extensive documentation on the package `isapnp`. This can be found on your system in the directory /usr/share/doc/packages/isapnp. At the end of this subsection we will take a closer look at configuring a PnP card, using the example of a sound card.

6.3 Sound Cards

Today, mainly PCI sound cards are available on the market. Sometimes you may also find ISA bus sound cards.

In SuSE Linux there are now two methods for supporting sound cards. The simplest method is to use YaST2. YaST2 uses drivers which were developed by the ALSA-project. Information on this project can be found at the URL: http://www.alsa-project.org. You also can install the ALSA-drivers with the program alsaconf.

6.3.1 Configuring Sound Cards with YaST2

To configure your sound card (if you didn't do this during the installation, e.g. because you added the sound card later to the system) just start YaST2 as user 'root' and choose the module 'Hardware/Sound'.

YaST2 now tries to automatically detect your sound card and brings up a list of drivers and corresponding sound card models suitable for your hardware. Select the driver and sound card model matching your card. Consult the sound card manual if necessary. If the sound card was not autodetected, click on 'Manual' and choose your sound card and the corresponding driver from the list. By selecting 'Options', you can enter some options specific for the selected card. The manual to your sound card gives more details.

By selecting 'Next' you will get to the "Test screen" to check the configuration of the sound card. The buttons '+' and '-' allow you to change the sound volume. You should start at a level of about 10% to avoid damage of your audio system or your ear. Also check the volume control of your audio system. By clicking on 'Test' you should hear a test sound. If there's no sound, you should check the volume of your audio system first. If you heard the sound and the volume is ok, click on 'Next'.

This finishes the configuration and the volume settings are saved. Click on 'Next' again to get back to 'YaST2 System configuration and administration'.

You can change the volume and mixer-settings later by using programs like kmix or gamix.

6.3.2 Configuring Your Sound Card With alsaconf

ALSA (Advanced Linux Sound Architecture) is supported by many developers and the number of drivers available for it is growing steadily. Along with a native interface, ALSA offers the OpenSoundSystem, an interface compatible to OSS. Thus you can continue to use applications that were originally developed for

OSS. All the sound configuration options can only be taken advantage of fully via the ALSA interface itself.

ALSA is completely based on the modular concept of the kernel. The required modules are located in the kernel module directory (`/lib/modules/...`) in the `misc` directory. Configuring ALSA takes place by means of entries in the `/etc/modules.conf` file. YaST2 takes care of creating these entries for you. However, ALSA offers its own program as well, alsaconf, which can be used in place of YaST2 for configuration.

With ALSA, you can run multiple sound cards simultaneously at any time. To do this, the `/etc/modules.conf` file only needs to be extended using the required parameters.

Proceed as follows to integrate your sound card into your SuSE Linux using alsaconf:

- Log in to a text console as user `root`.
- Start the program alsaconf

  ```
  earth:/ # alsaconf
  ```

- alsaconf will at first attempt to detect the sound card. It not only searches for ISA bus cards, but also for PCI bus cards. Plug and Play cards on the ISA bus are scanned as well.
- alsaconf shows you a list of the sound cards it found and offers a configuration for them. If a card has not been found, a selection list of potential cards will immediately appear.
- Now you will be asked whether the `/etc/modules.conf` file should be modified. You should confirm this by selecting `YES`.
- The necessary entries are now created in `/etc/modules.conf` and the driver will be given a trial run.
- Subsequently, you have the option of testing the sound card. A “test sound” will be played back for this purpose. Do not turn the speakers up too loud.
- The drivers will automatically be loaded the next time you start your system. You can also manually load and unload them.

  Enter

  ```
  earth:/ # rcalsasound start
  ```

  to load and

  ```
  earth:/ # rcalsasound stop
  ```

  to unload the drivers.

6.4 Ports on a Computer

6.4.1 PS/2 Ports

PS/2 ports were developed by IBM for the PS/2 platform. The ATX Standard, in which this type of port is included, has found widespread use in the last few years. You can attach a mouse and a keyboard on the PS/2 port.

Configuring the PS/2 port is not necessary under Linux because it is supported by the kernel. The system automatically detects existing PS/2 ports and can control the devices attached to them. The ports can be found and addressed under the device name `/dev/kbd` and `/dev/psaux`.

6.4.2 Serial Ports

The serial port (RS232) of your computer is often used for external modems. You can also attach a mouse with an RS232 plug on this port.

For the support of this port in Linux, installation of the kernel module `serial.o` is required. This module is automatically loaded when the port is addressed by certain software. For instance, supposing you've configured your Internet access such that a modem on `COM1` is used, the module is automatically post-installed before the connection to your provider is established.

You can change your serial port configuration using the program setserial. This program is run at every system start by the script `/sbin/init.d/serial`. For more information on setserial see the manpage for this program. Serial ports can be found and addressed under the device file(s) `/dev/ttyS0`, `/dev/ttyS1`, etc.

6.4.3 Parallel Ports

The parallel port of your computer is primarily used to connect printers. Many other devices such as ZIP-Drives and CD-ROM/RAM Drives can also be connected to this port.

The "parport" subsystem of the current Linux-kernels 2.4.x is designed to operate a number of devices on the parallel port *simultaneously*. As an example, you can attach a ZIP drive to the parallel port and, in turn, connect a printer to the ZIP drive.

Initializing `parport`

The `parport` and `parport_pc` kernel support are required for this. `parport` is the actual subsystem of the kernel and `parport_pc` takes care of the hardware connections of the PC port as well as for those of some AXP computers. With the command

```
earth: # modprobe parport_pc
```

you can attach the modules manually. Check if the ports could be found and configured by entering the command **`lsmod`**; the `parport` and `parport_pc` ought to appear in the list of modules. You can also look in the file `/var/log/messages` to check which values were used to initialize `parport` (see File contents 6.4.1). The entries in the file 6.4.2 are responsible for configuring the

```
Jun  3 09:15:53 tux kernel: parport0: PC-style at 0x378 [SPP,
                                                          ECP]
```

File contents 6.4.1: `/var/log/messages`: `parport` initialization

interfaces; the entry which is commented out indicates how to integrate another parallel interface.

```
alias parport_lowlevel    parport_pc
options parport_pc io=0x378 irq=none

# If you have multiple parallel ports, specify them this way:
# options parport_pc io=0x378,0x278  irq=none,none
```

File contents 6.4.2: `/etc/modules.conf`: `parport` configuration

> **Note**
>
> In case of problems, you should first check in the BIOS of the computer how the port in question is specified there. The port should have its own explicit address (if possible, `0x378`) and its own interrupt. “Automatic” settings should be avoided.

Sometimes it is also necessary to switch on the devices in the correct sequence; please look in the appropriate handbooks! One recommended method is to first switch on the computer, and then – before booting has commenced – to *immediately* switch on the external devices.

Using `parport`: Printers, ZIP Drives, PLIP, etc.

If one parallel port device is to be used, its relevant module must also be loaded; in the case of a printer, this is the `lp` module:

```
earth: # modprobe lp
```

`lsmod` ought to show us, amongst other things:

```
parport_pc              5568   1  (autoclean)
parport                 6884   1  [parport_pc lp]
lp                      5116   0  (unused)
```

As with other cases, further details are supplied in the file `/var/log/messages` and also in the `proc` file system in `/proc/parport`. If everything is running correctly, you won’t need to load the modules manually – Kmod will do this for you automatically (Section 9.2 page 239) if, for example, a printer job is sent. ZIP-drives are not served either by `ppa` or by `imm` – see below, Section 6.5.4 page 185.

Further Suggestions

Look in the kernel sources, in directory `/usr/src/linux/Documentation`, at the files `parport.txt` and `paride.txt`.

Information on the current status of `parport` programming can be found on the web at `http://www.torque.net/linux-pp.html` and `http://www.torque.net/parport/`.

6.4.4 USB – Universal Serial Bus

In 1994 Compaq, Intel, Microsoft and NEC cooperated to setup the specifications for the "Universal Serial Bus" (USB). They aimed to create an easy-to-use connection to the telephone and an extension of the already available connection types (serial, parallel). Using this bus system, you are able to connect various devices like mice, keyboards, printers, scanners and joysticks to your computer. The different cabling used to connect these devices should be replaced by one , unified connection type. The USB bus is capable of mounting and unmounting devices to your computer while it is running – the computer does not need to be rebooted.

The first version of the USB bus was released in 1996 and extended it's capabilities from release to release. In the current version (USB 1.1) 127 devices are supported. The maximum bandwidth is limited to 12 Mbits/s.

In Linux, support for the USB bus is constantly growing. Numerous devices are already supported and can be configured easily. From kernel version 2.4 on, the usage of USB devices with Linux will be standard.

Available Modules

In the following section we are going to give you an overview on the available modules and will give you details which module should be loaded for which device. The listed modules are located in the directory `/lib/modules/xxx/usb`. Details on the modules can be found in the directory `/usr/src/linux/Documention/usb/`.

If we talk about "input events" or "events" in general here, you should think about mouse movements, keystrokes, etc.

input.o This module needs to be loaded before other USB input devices can be supported by Linux. It is used to control the communication between the hardware (via USB) and the processing of the events. More information on that can be found in the file `input.txt` in the directory `/usr/src/linux/Documentation/usb`.

usbcore.o USB core is the USB-subsystem that simplifies the communication with the installed hardware which depends on the data structures, macros and functions of the hardware. By using a general interface to the operating system (API) common routines of all USB devices and host controllers are supported.

usb-uhci.o / usb-ohci.o This module enables the USB host controller of your motherboard. Most of the motherboards available today have the USB host controller integrated in the chipset. Older motherboards can be upgraded by a PCI card. The USB host controller is then integrated on this PCI card. There are two standards making the USB host controllers compatible: The Universal Host Controller Interface (UHCI) Intel and the Open Host Controller Interface (OHCI) defined by Compaq.

The module `usb-ohci.o` enables the OHCI-chipset (Compaq, NEC, iMacs, OPTi, SiS, ALi, Lucent, CMD, etc.) whereas the `usb-uhci.o` module enables the UHCI-chipset (Intel, VIA) controllers.

Tip

If you are logged in as user **`root`**, you are able to read out the type of USB host controller on your motherboard by using the command **`lspci -v | grep USB`**. You can then load the respective module.

hid.o HID (Human Interface Devices) is a device class accessed by this generic module. Supported devices are keyboards, mice, joysticks, digitizers etc.

For some devices there are two modules available in the modules directory: The `*dev.o` module is a “high-layer”-module (“*” standing for e.g. “mouse”) and is meant to as an interface between the hardware and the users program like “X” or “games”; `usb*.o` is a “low-layer”-module which generates the input events. Both modules are needed, whereas the “low-layer”-module can be replaced by the `hid.o` module.

There are two modes to describe the event input of a USB device: *HID* enables support of all functions of a device so requires the quite complex module `hid.o`: *HIDBP* (HID boot protocol) is a very simple mode and only supports the basic functionality of a device to make it accessible by the computers BIOS.

usbkbd.o This module enables the usage of the basic functions of a keyboard in the HIDBP-mode, but for this device class the access should be enabled via the `hid.o` module. Unlike the `hid.o` module, only keyboards will be enabled with this module.

usbmouse.o This module enables using the basic functionality of a mouse in HIDBP-mode. The usage of the `hid.o` should be preferred as this enables event management for all other input devices as well.

keydev.o This module “translates” input events to the architecture dependent RAW mode of the keyboard. This module is still under development but works fine for almost every architecture.

mousedev.o This module converts the input events of a mouse, digitizer or tablet to PS/2 compatible events. As it is not finished yet, you should use the `evdev.o` module or, if you are using a “wheel”-mouse, the `hid.o` module.

acm.o Load this module if you plan to use communication devices like modems or ISDN adapters of the device class “ACM” (Abstract Control Model). “ACM” is a method to control modems. You can find a list of supported modems in the file `acm.txt` in the directory `/usr/src/linux/Documentation/usb`.

audio.o enables audio devices and supports digital mixers, wave devices and loudspeakers.

dabusb.o supports amateur radio devices.

dc2xx.o enables digital cameras of the *Kodak DC-2xx* series and also some non-*Kodak*-cameras. You can find more information on this in the file `dc2xx.txt` in the directory `/usr/src/linux/Documentation/usb`.

evdev.o “evdev” is a general interface (API) controlling the input of devices as event codes are identical and hardware-independent on all architectures. This module supports this API.

ibmcam.o is used for supporting the *IBM*(C-It) and *Xirlink*-Webcams, but not all features are supported now. You may find more details on that in the file `ibmcam.txt` in the directory `/usr/src/linux/Documentation/usb`.

joydev.o is used to communicate with joysticks.

ov511.o enables the “OV511” chip from *Omnivision Techonolgies* which is used in some webcams. In addition, the *OV511/OV511+*- and *OV7610/20/20AE CCD*-chips are supported. This driver is compatible with most “Video4Linux” applications, but not to some video conferencing applications. For details on the supported resolutions, please read `ov511.txt` in the directory `/usr/src/linux/Documentation/usb`.

printer.o is used to communicate with USB printers.

scanner.o was developed to support USB scanners from *Hewlett Packard* (HP) first, but became more and more hardware independant later.
Read more about it in `input.txt` in the directory `/usr/src/linux/Documentation/usb`.

usb-serial.o With this module you can control serial devices on the USB bus. The serial signal are converted to USB signals. Up to 256 connections can be used simultaneously. Details can be found in the file `usb-serial.txt` in the directory `/usr/src/linux/Documentation/usb`.

usb-storage.o enables USB mass storage devices like ZIP, floppy disk drives, hard disks and smart media card devices. The driver isn’t quite stable at the moment.

wacom.o enables support for the *Wacom Graphire and Intuos Tablets*. Although support should already be included with the *HID*-module, it was only possible to use a *Wacom PenPartner*.

wmforce.o enables the use of *Logitech WingMan Force* joysticks with USB-interface. The driver works well, but the “Force Feedback” function is not yet supported.

Configuring a USB mouse and keyboard

For a common configuration with USB mouse and keyboard the following modules should be loaded:

- input.o
- mousedev.o
- keybdev.o
- usbcore.o
- usb-uhci.o or usb-ohci.o
- hid.o

Further sources

More information on the USB-implementation and on supported devices can be found at

- `http://www.linux-usb.org`
- `http://www.qbik.ch/usb/devices/`

6.5 Removable Drives

A variety of removable drives can be used under Linux: Floppy Disk Drives, ZIP-, JAZ- or SyQuest Drives. Magnetic Optical drives can

6.5.1 Floppy Disk Drives

The package `mtools` was developed for easy access of MS-DOS formatted floppy disks.

You can also read and write to disks formatted with non-MS-DOS file systems (e.g. `ext2` or `minix`). Many other file systems are also available. In order to accomplish this, however, you’ll first need to use the command mount to access the medium.

6.5.2 LS-120 Drives

LS-120 drives are connected at the (E)IDE port of the computer and are seen by the system as hard disks, meaning that they, too, need to be accessed first with the command mount.

6.5.3 MO Drives

MO drives (engl. *Magneto-Optical Disks*) are supported by Linux, as long as the SCSI controller connected to the drive can be used with Linux. This medium is then accessed just as the SCSI hard disks are accessed; therefore, it is necessary to “mount” the medium.

6.5.4 ZIP Drives

There are ZIP drives for different interfaces: parallel, ATAPI and SCSI, as well as USB.

For IDE and SCSI, you don’t need a special device driver. They can be connected on the bus and addressed without extra configuration using the bus’s device driver (IDE or SCSI). The devices are addressed by the device names,

```
/dev/hda - /dev/hdd for IDE (ATAPI)
```

and

```
/dev/sda - /dev/sdm for SCSI
```

The device needs to be tied into the directory tree with the command mount.

The installation of the *parallel port* version proves to be somewhat more complicated. You’ll need SCSI hard disk support, `parport`, `parport_pc` support (see Section 6.4.3 page 179) as well as `ppa` or the `imm` driver of the kernel. [1] `imm` is needed for more modern drives (e. g. for the ZIP 250). Furthermore you should make sure that, in the BIOS of the computer, the EPP mode for the parallel port is set. If you are unsure as to which kind of drive you have, try loading `imm`:

```
  earth: # modprobe imm
```

If this doesn’t work try the same with `ppa`:

```
  earth: # modprobe ppa
```

With this, the `parport` subsystem will be automatically initialized. If this does not function, look at Section 6.4.3 page 179.

Sometimes an entry in `/etc/modules.conf` can help matters (see File contents 6.5.1 on the next page). The `alias scsi_hostadapter` may only be used if you don’t have a standard SCSI host adapter in your system. An alternative is to write the necessary **`modprobe`** commands into the boot script `/sbin/init.d/boot.local`; For more information on this file, see Section 13.4 page 313.

[1] `ppa`- and/or the `imm` driver are contained in the “SCSI low-level drivers” – but you *don’t* need to compile your own kernel; the necessary modules are pre-compiled and included with SuSE Linux

```
# alias scsi_hostadapter ppa
pre-install ppa modprobe "-k" parport_pc
```

File contents 6.5.1: /etc/modules.conf: ppa-configuration

After this, ZIP disks can be accessed in the same manner as SCSI hard disks. It is also necessary to “mount” the medium.

Adding support for USB ZIP drives is quite easy to manage. First, make sure that the USB subsystem is loaded (see Section 6.4.4 page 181). The following modules need to be loaded:

```
usbcore
usb-uhci
```

or

```
usbcore
usb-ohci
```

In addition you have to load the modules named usb-storage.

Then you can access the ZIP drive just like SCSI hard drive. You need to “mount” the media first.

6.6 Modems

6.6.1 External Modems

External modems are normally attached directly to the serial port of the computer. Programs can address them via device files with the name(s) /dev/ttyS0, /dev/ttyS1. etc.

6.6.2 Internal Modems

Internal modems in desktop machines are either PCI or ISA devices. Please see Section 6.2 page 173 for information about these devices.

Since modems are accessed through serial ports, internal modems must make these ports available for themselves.

Table 6.1 gives you a list of standard resources for serial ports.

Device file	Name	Interrupt	IO-Address
/dev/ttyS0	COM1	4	0x3f8
/dev/ttyS1	COM2	3	0x2f8
/dev/ttyS2	COM3	4	0x3e8
/dev/ttyS3	COM4	3	0x2e8

Table 6.1: Standard resources for serial ports

PCI modems

If you have a PCI modem, it is most likely a “Winmodem”. Winmodems are not modems. They are hardware devices which need a special driver that allows them to emulate a real modem. This driver is generally written by the makers of the Winmodem and currently only runs on Microsoft Windows.

You can obtain current information about Linux-compatible modems and the development of Winmodem drivers at: `http://www.o2.net/~gromitkc/winmodem.html`.

If the modem is not a Winmodem, you should be able to use it in SuSE Linux. A certain amount of manual work is necessary, however, since these modems cannot be configured automatically.

ISA modems

With ISA modems, there are two main considerations:

- The modem itself needs to be configured (IRQ and IO addresses for the port).
- The kernel needs to know what IRQ and port the card is using.

Initializing the modem

As described in Section 6.2 page 173, ISA cards can be jumpered, jumperless or Plug and Play. The initializing of PNP cards is also described in that section.

Some suggestions about settings:

Setting your modem to use `/dev/ttyS1` (`COM2`) with IRQ 3 will usually present the least number of problems. `/dev/ttyS3` (IRQ 4) can also be used, but you may have a conflict if you are also using ttyS0 (`COM1`), as this port also normally uses IRQ 4. So if you have the mouse on /dev/ttyS0 you should not use `/dev/ttyS0` or `/dev/ttyS2`.

Generally, IRQ 5 and IRQ 7 should also be avoided, as these are often used for sound cards or printers. In fact, if you are going to use a sound card, you should configure it first because it claims many resources.

Passing parameters to the kernel

After you have established the port and IRQ for your modem, you need to tell the kernel what values the card is using. If you have used a standard port and IRQ (`/dev/ttyS0` or `/dev/ttyS1`), your card will be automatically recognized the next time you reboot.

If you are using other IO-port/IRQ combinations, you can manually tell the kernel the device’s IRQ, using setserial:

```
earth:/ # setserial /dev/ttyS3 irq 10
```

You can now configure the modem for Internet connection.

If the modem is recognized, you can automate the setserial command by modifying the appropriate line in the file `/sbin/init.d/serial`:

Change the line:

```
# run_setserial /dev/ttyS3 $AUTO_IRQ autoconfig
```

to:

```
run_setserial /dev/ttyS3 irq 10
```

6.7 Scanners

In order to use a scanner in Linux, as well as any other operating system, a collection of special drivers and programs is needed. The package `sane` (series `gra`) offers such a collection. With the help of SANE you can, in Linux, use a scanner that is connected to a kernel-supported SCSI adapter.

Scanners that are connected to the parallel port (printer port) of the computer are not yet supported although device drivers are in development as for USB.

Because SANE is in continual development the list of supported scanners is growing. For a list of the latest supported scanners go to `http://www.mostang.com/sane/sane-backends.html`.

How Do You Configure a Scanner In Linux?

Scanners are addressed as “generic scsi devices” in Linux. The respective device names are: `/dev/sg0`, `/dev/sg1` ...etc.

To find the matching file in your system, use the program sgcheck:

```
earth:/ # sgcheck
```

You should receive the following output:

```
Assignment of generic SCSI devices,
device host/channel/ID/LUN type(numeric type) vendor model:

        /dev/sg0 0/0/0/0 Direct-Access(0) SEAGATE ST32550N
        /dev/sg1 0/0/1/0 CD-ROM(5) PIONEER CD-ROM DR-U10X
        /dev/sg2 0/0/5/0 Processor(3) HP C2500A
```

Scanners have a device description like `Processor` or `SCANNER`. In the example above you could address the scanner with the device name `/dev/sg2`.

You need to make a symbolic link to the device to `/dev/scanner`:

```
earth:/ # ln -s /dev/sg2 /dev/scanner
```

Finally you need to change the permissions for the generic SCSI device. SANE needs read as well as write permissions for the device (it sends commands to the scanner as well as receiving information):

```
earth:/ # chmod 777 /dev/sg2
```

Problems

If SANE doesn't find your scanner, check first to make sure that your SCSI controller is found and correctly configured:

```
earth:/ # cat /proc/scsi/scsi
```

You should receive output that resembles the following:

```
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
 Vendor: SEAGATE Model:  ST32550N         Rev: 0016
 Type:   Direct Access                    ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 01 Lun: 00
 Vendor: PIONEER Model:  CD-ROM DR-U10X   Rev: 1.07
 Type:   CD-ROM                           ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 02 Lun: 00
 Vendor: QUANTUM Model:  FIREBALL TM3200S Rev: 1.07
 Type:   Direct-Access                    ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 06 Lun: 00
 Vendor: SCANNER Model:                   Rev: 2.00
 Type:   Scanner                       ANSI SCSI revision: 01 CCS
```

If you can't find an entry like "SCANNER" or "PROCESSOR", your scanner hasn't been identified by the SCSI controller. Check to see that the scanner is switched on and that it, as well as the SCSI bus itself, are correctly attached.

6.8 Tape Drives

Tape drives, also known as streamers, are mainly used for high-capacity backup purposes. There are tape drives on the market for a variety of ports.

SCSI Tape Drives

Tape drives that attach to the SCSI bus are the most common. You can assume that all devices that are attached to a Linux-supported SCSI controller can be used, although if your device has extra functions, special software is needed to access these.

A SCSI tape drive is controlled in the following way:

- The first tape drive in the system is controlled using the device files `/dev/st0` and `/dev/nst0`, the next with `/dev/st1` and `/dev/nst1`, etc. `/dev/stX` stands for a drive that automatically rewinds after use. `/dev/nstX` on the other hand stands for a tape drive that leaves the tape in the current position after use.

  We suggest using `/dev/nstX`, because, using mt, you have exact control of the drive.

- Use the command mt to control a tape drive:

  ```
  earth:/ # mt -f /dev/nst0 retension
  ```

  Adjusts the tape tension by rewinding and then fast-forwarding the tape.

```
earth:/ # mt -f /dev/nst0 rewind
```

Moves the tape to a point just after the last recorded information. Used to add information to a partially used tape.

```
earth:/ # mt -f /dev/nst0 eof
```

Moves the tape to a point just after the last recorded information. Used to add information to a partially used tape.

```
earth:/ # mt -f /dev/nst0 erase
```

Erases all information on a tape. Be careful, there is no way to recover erased information!

IDE Tape Drives

IDE tape drives are connected to the IDE bus of the computer. The drive is found automatically by Linux. You can address the device under the device files `/dev/ht0` or `/dev/nht0`.

You can control the device using the command mt, as given in Section 6.8 on the page before.

Floppy Tape Drives

Floppy tape drives are simply attached to the floppy disk port of your computer (almost every computer has a floppy port).

Because these devices need to be controlled with precise timing, errors often occur while saving and writing backups.

To use these devices in Linux, you'll need a special driver, the ftape driver. This driver is included in the kernel. Read the documentation in `/usr/src/linux/Documentation/ftape.txt` for more information on the configuration of this device. The entire documentation for the ftape subsystem can be found on the device driver developers' homepage: `http://www.instmath.rwth-aachen.de/~heine/ftape/`.

7 Notebooks – PCMCIA, APM, IrDA

Notebooks, in particular, have special components and requirements, among these are "Advanced Power Management" (APM), Infrared ports (IrDA) and PC cards (PCMCIA). Occasionally these components can also be found in desktop computers. Because the differences between the two types are insignificant, the use and configuration of both will be described in this chapter.

Those interested in specific notebooks should visit the "Linux laptop homepage" at `http://www.cs.utexas.edu/users/kharker/linux-laptop`. Another good source of information is the "Moblix" homepage at `http://mobilix.org/index.html` (*MobiliX – Mobile Computers and Unix*). There you can find an interesting Laptop HOWTO as well as an IrDA HOWTO. In addition, there is an article *Laptops and Notebooks (PCMCIA) in Linux* (`file:/usr/share/doc/sdb/de/html/laptop.html`) in the SuSE support database.

7.1 PCMCIA

7.1.1 Hardware

PCMCIA stands for "Personal Computer Memory Card International Association" although it is commonly used as a collective term for both the hardware and the associated software. The essential part is the PCMCIA card, of which two types exist:

PC cards: This is the most common type. With a 16-bit bus bandwidth, most are relatively inexpensive and are usually trouble-free and stable.

CardBus cards: This is a new standard. They have a 32-bit bus bandwidth and are quicker, although also more expensive. Because the data transfer rate is often limited at other points in the system, the increase in bandwidth is often unjustified. In the meantime there are quite a few drivers for these cards as well, although they are often unstable – depending on the PCMCIA controller.

When the PCMCIA service is active, the card type can be found with the command **`cardctl ident`**. A list of supported cards can be found in `SUPPORTED_CARDS` in `/usr/share/doc/packages/pcmcia`. There you'll also find the actualized version of the `PCMCIA-HOWTO`.

The second important component is the PCMCIA controller, or the PC card or CardBus bridge. This creates a connection between the card itself and the PCI bus or, in older devices, the ISA bus. These controllers are most often compatible to the `i82365` chip from Intel; all models are supported. The type of controller can be found with the command **`probe`**. In the case of PCI devices, the command **`lspci -vt`** also offers interesting information.

7.1.2 Software

All necessary drivers and programs can be found, as long as they are not already integrated into the kernel, in package `PCMCIA`, series `a1`. The modules `pcmcia_core`, `i82365` (or `tcic`, less often) and `ds` form the basis and are normally automatically started when booting. They initialize the PCMCIA controller and supply basic functions.

Because PCMCIA cards can be added or removed at run time, a daemon is needed to check the activities of the slots. This is done by the Cardmanager (**`cardmgr`**), which is automatically started after loading the base modules. When a card is inserted into a slot Cardmanager determines the type and function and loads the necessary module. You can check which modules are loaded with the command **`lsmod`**. If all modules are successfully loaded Cardmanager, depending on the function of the card, starts predetermined initialization scripts that build the necessary network connections or *mount* the necessary external SCSI drives. When the card is removed Cardmanager, using the same scripts, ends the diverse card activities, after which the modules that are no longer needed are unloaded. Theoretically, you can simply remove the card. This works very well with network, modem, or ISDN cards as long as there is no active network connection. It does not work with mounted partitions, external drives, or NFS directories. For this you need to be certain that the devices are synchronized and cleanly *unmounted*. If in doubt the following command can be of help

```
earth: # cardctl eject
```

This command deactivates the card, as long as it is still in the notebook.

7.1.3 Configuration

When the PCMCIA package is installed, four PCMCIA variables will be found in `/etc/rc.config`. **`START_PCMCIA`** determines whether the service should be started when booting. It can be started with the command **`rcpcmcia start`** even when **`START_PCMCIA`** is set to `no`. **`PCMCIA`** determines the type of PCMCIA controller. This value is automatically set at installation and almost always has the value `i82365`. The other two variables, **`PCMCIA_PCIC_OPTS`** and **`PCMCIA_CORE_OPTS`** involve options for the basis modules and can normally be left empty.

Because the selection of driver modules is taken care of by Cardmanager (**`cardmgr`**) no other hardware related settings are necessary.

Ethernet and Token Ring

Ethernet or token ring connections can be easily configured using YaST. In the menu item '`System administration`' → '`Network configuration`', these devices can be configured in the same manner as normal hardware devices with the exception that the device must be marked as a PCMCIA device with F9; see also Section 3.6.3 page 83. The following points should be taken into consideration:

- When more than one ethernet or token ring device is marked as PCMCIA, only the first device configuration is used.
- The settings, in contrast to normal network cards, are immediately activated with `F10`.
- The device names shown (`eth0`, `eth1`, `tr0...`) should only be seen as information relating to the type of device because the numeration of PCMCIA devices is determined dynamically.
- When a non-active device is reactivated with `F4` the device must also be marked as a PCMCIA device with `F9` again.

ISDN

ISDN-PC cards are configured in much the same manner as other cards. The only difference is that the device must be marked as a PCMCIA device under the menu items '`System administration`' → '`Network configuration`' with `F9`. In the '`Configuration of ISDN hardware`' dialog, IRQ, IO port, and the ISDN protocol (Euro-ISDN or 1TR6) should not be changed.

ISDN modems are also available as PCMCIA cards. They are modem or multi-functional cards with an additional "ISDN Connection Kit" and are handled as modems.

Modems

Modem PC cards normally have no specific settings. After insertion, the card is available in `/dev/modem`. The configuration of the device with wvdial (**`wvdial`**) follows that of a normal modem.

SCSI and IDE

The necessary driver modules are loaded by Cardmanager. After insertion, the device is ready for use. The device name is determined dynamically. Information about available SCSI or IDE devices can be found under `/proc/scsi` or `/proc/ide`.

> **Note**
>
> Before inserting external hard disks, CD-ROM drives and similar devices be sure that the device is on. SCSI devices must be actively terminated. *Important:* Before a SCSI or IDE device is removed from the slot, the partitions or device must be unmounted. If this step is left out, the device can only be accessed after rebooting although the system is still stable.

It is possible to install Linux entirely on such an external drive, but the boot process is somewhat more complicated. A "boot disk" will be needed which includes the kernel and an init RAM disk (`initrd`); more information on this can be found in Section 12.4 page 294. The `initrd` includes a virtual file system with all the necessary PCMCIA modules and programs. The SuSE Linux “boot disk” and boot disk images include these, enabling booting of your external installation. It is, however, somewhat uncomfortable having to load the PCMCIA support every time you boot your system; advanced users can create their own boot disks that are tailored to their systems[1].

7.1.4 Configurations for Changing – “Schemes”

Mobile computers often have different configurations. For example, one for work and one for home. For PCMCIA devices this proves to be quite easy, although the configuration files need to be edited by hand; YaST cannot yet handle this task at the moment . [2].

Singular configuration profiles for PCMCIA are known as “Schemes”. The configuration files found in `/etc/pcmcia/*.opts` can include more than one scheme. The data for a specific scheme is included in configuration blocks and selected with an “address”. The addresses are comma-separated words. The first word is the name of the scheme. The other words contain information such as the slot in which the card is inserted or the number of a hard disk partition. Detailed information can be found at the beginning of the `/etc/pcmcia/*.opts` files as well as in the PCMCIA-HOWTO. The standard scheme used by SuSE Linux has the name `SuSE`. The `SuSE` scheme is edited with SuSEconfig, when the network connection is configured with YaST. Due to this, manual alterations to this scheme will be lost by using **`SuSEconfig`**.

We’ll use an ethernet card configuration as an example scheme configuration, making two schemes named `work` and `home`. The file `/etc/pcmcia/network.opts` (see file 7.1.1 on the facing page) will have several configuration blocks with the following addresses:

- `SuSE,*,*,*`: This block already exists and should be left as is, in case you’d like to use YaST for future configurations.
- `work,*,*,*`: This block is for the network configuration on the job.
- `home,*,*,*`: The configuration for home.
- `*,*,*,*`: This block already exists as well and should be left as is for use in future configurations.

It is easiest to use YaST to configure a scheme and then change the name, after running **`SuSEconfig`**, from `SuSE` to `work` or `home`. Those who wish to edit everything themselves can find more information about terms such as **`NETMASK`**, **`BROADCAST`**, etc., and all specific values in `PCMCIA-HOWTO` (in `/usr/share/doc/packages/pcmcia` you’ll find the current version).

[1] Tips can be found in the PCMCIA-HOWTO in Section 5.3 “Booting from a PCMCIA device.”

[2] Several ISDN devices, with the same network address and the same default route, can be configured. These are not started automatically, but selectively. In the case of modems, several different profiles can be configured with the wvdial

```
# The address format is "scheme,socket,instance,hwaddr" .
case "$ADDRESS" in
SuSE,*,*,*)
    INFO="This scheme is to be configured by YaST/SuSEconfig"
# [... abbreviated ...]
    ;;
work,*,*,*)
    INFO="Network configuration for the company via DHCP"
    IF_PORT=""
    BOOTP="n"
    DHCP="y"
    IPADDR=""
    NETMASK=""
    NETWORK=""
    BROADCAST=""
    GATEWAY=""
    DOMAIN=""
    SEARCH=""
    DNS_1=""
    MOUNTS=""
    start_fn () { return; }
    stop_fn () { return; }
    ;;
home,*,*,*)
    INFO="Network configuration for home, using a fixed address"
    IF_PORT=""
    BOOTP="n"
    DHCP="n"
    IPADDR="10.0.1.23"
    NETMASK="255.255.255.0"
    NETWORK="10.0.1.0"
    BROADCAST="10.0.1.255"
    GATEWAY="10.0.1.1"
    DOMAIN="home.de"
    SEARCH="home.de work.de"
    DNS_1="10.0.1.1"
    MOUNTS=""
    start_fn () { return; }
    stop_fn () { return; }
    ;;
*,*,*,*)
    INFO="Sample private network setup"
# [... abbreviated ...]
    ;;
esac
```

File contents 7.1.1: /etc/pcmcia/network.opts

```
image    = /boot/vmlinuz
  root   = /dev/hda7
  label  = work
  alias  = w
  append = xzx SCHEME=work xzx
#
image    = /boot/vmlinuz
  root   = /dev/hda7
  label  = home
  alias  = h
  append = xzx SCHEME=home xzx
#
image    = /boot/vmlinuz
  root   = /dev/hda7
  label  = suseconf
  alias  = s
  append = xzx SCHEME=SuSE xzx
```

File contents 7.1.2: PCMCIA: example from `lilo.conf`

Scheme changes can be made at boot or run time. The active scheme is defined by the command **`cardctl scheme`**. The changes can be made at run time with the same command using the syntax **`cardctl scheme`** <**`Scheme_name`**>. These settings are not lost when restarting the system, which means that the last selected scheme is still active at the next boot unless specifically changed.

This is done either in the '`boot options`' or directly at the boot prompt (**`LILO:`**) by adding the following variable to the command: **`SCHEME=`**<**`Scheme_name`**>; more information on giving parameters at the boot prompt can be found in Section 10.3.2 page 247.

You can use the boot manager to create one or more fixed boot schemes. We will continue with our example and create three boot configurations. The **`append`** parameter in `/etc/lilo.conf` is used for this and other purposes as can be seen in File 7.1.2. After reconfiguring `/etc/lilo.conf` the command **`lilo`** needs to be called.

Using the example configuration, in the file 7.1.2 by simply adding `w`, `h` or `s` when booting, a specific scheme can be started.

7.1.5 If Things Still Don't Work

Some notebooks have problems with specific PCMCIA cards. Most can be easily solved when you look at the matter systematically. First check to see if the problem lies with the card or the PCMCIA base system. To do this, boot the computer *without* any cards inserted. Only after the base system functions correctly should you insert a card. All important messages can be found in `/var/log/messages`. Therefore this information should be observed with

```
earth: # tail -f /var/log/messages
```

while the necessary tests are running. Doing so can reduce the problem to one of the following two errors.

The PCMCIA base system does not function properly

If the system hangs when booting, after you receive the message `"PCMCIA: Starting services:"` or other strange things happen, you can prevent PCMCIA from starting at the next boot by entering the parameter **`NOPCMCIA=yes`** at the boot prompt (**`LILO:`**) . After the system is running you should load the modules, one after the other, manually, with the commands **`modprobe pcmcia_core`**, **`modprobe i82365`** or – in very rare circumstances – **`modprobe tcic`** and **`modprobe ds`**. The critical modules are, in each case, the first ones.

If the problem occurs when loading `pcmcia_core` help can be found in manpage for **`pcmcia_core`** (**`man pcmcia_core`**). The options described in this can be tested in conjunction with the **`modprobe`** command.

As an example, we can shut off the APM support of the PCMCIA module; in rare instances, this can lead to difficulties. For this, we can use the option **`do_apm`**; via **`do_apm=0`**, the power management will be de-activated:

```
earth:~ # modprobe pcmcia_core do_apm=0
```

If this option is successful, you should set the variable **`PCMCIA_CORE_OPTS`** in `/etc/rc.config` to:

```
PCMCIA_CORE_OPTS="do_apm=0"
```

Also, from time to time, verifying free IO regions can cause some distress if other hardware components are disturbed by it. You can bypass this via **`probe_io=0`**. If you want to use a number of options, they should be separated by a space:

```
PCMCIA_CORE_OPTS="do_apm=0 probe_io=0"
```

If the problem occurs when loading `i82365`, help can be found in manpage for **`i82365`** (**`man i82365`**).

This error is the result of a resource conflict, which means that an interrupt, an IO port, or a memory area is being shared by two devices. The module i82365 checks for such sharing but sometimes this check itself leads to problems. Checking the interrupt 12 (PS/2 devices), for example, can lead, on some computers, to the mouse or keyboard being blocked. In this case the parameter **`irq_list=<List_of_IRQs>`** helps. This list should include all interrupts to be used by the base system and has the following syntax:

```
earth: # modprobe i82365 irq_list=5,7,9,10
```

or permanently in `/etc/rc.config`:

```
PCMCIA_PCIC_OPTS="irq_list=5,7,9,10"
```

In addition to this there are the files `/etc/pcmcia/config` and `/etc/pcmcia/config.opts` which are used by Cardmanager. The settings in these files are used for the loading of driver modules for the PCMCIA cards. IRQs, IO-Ports and memory areas can be assigned or unassigned in the file `/etc/pcmcia/config.opts` as well. The difference between this and the above option is that when resources are unassigned in this file the PCMCIA card does not use them, but the base system module `i82365` still checks them.

The PCMCIA card doesn't function (correctly)

There are three possibilities for errors: the card is not correctly detected, the card causes a resource conflict, or the card is not supported.

If the card is not properly detected, the message `"unsupported card in Slot x"` will appear in `/var/log/messages`. This message only means that Cardmanager cannot correctly assign a driver. `/etc/pcmcia/config` is used for this purpose; it is, so to speak, a "driver database". You can add entries to the "driver database" using existing entries as an example. Using the command **`cardctl ident`**, you can determine the identification information for the card. More information on this topic can be found in the PCMCIA-HOWTO, Section 6 ("Dealing with unsupported cards") as well as the manpage for **`pcmcia`** (**`man pcmcia`**). After changing `/etc/pcmcia/config` be sure to reload it with **`rcpcmcia reload`**.

Generally speaking, it doesn't matter which IRQ or IO port a PCMCIA card uses as long as these are not shared by another device (CD-ROM on 2. IDE-Controller: IRQ 15; serial or IrDA Port: IRQ 3, IRQ 4; sound, printer: IRQ 5, IRQ 7). In cases of resource conflicts, these should be specifically defined in `/etc/pcmcia/config.opts`. If the problem still exists, these settings can be further specified as a module option in `config.opts`. For example, if the module `pcnet_cs` should use IRQ 5 the following entry would be made:

```
module "pcnet_cs" opts "irq_list=5"
```

Most available options are described in the manpages[3]. If the manpage does not include the information needed, there are several options available in manpage for **`i82365`** (**`man i82365`**) or the answer lies in the source code of the module.

One common problem with 10/100 Mbit-Network cards is incorrect detection of the transfer rate. In this case the command **`ifport`** can be of help. With this the transfer rate can be shown and changed; see also manpage for **`ifport`** (**`man ifport`**). The correct setting can be entered in the variable **`IFPORT`** in `/etc/pcmcia/network.opts`.

7.1.6 Installation via PCMCIA

In some cases PCMCIA support is needed to install SuSE Linux. For this you should choose '`Load PCMCIA Module`' from the menu item '`Kernel-Module (Hardware Drivers)`' in linuxrc. At first two entry fields will appear. Here you can specify options for the modules `pcmcia_core` and `i82365`. Normally these can be left empty. The manpages for `pcmcia_core` and `i82365` can be found as text files on the first CD in the directory `docu`. During installation, system messages are displayed on several virtual consoles and can be viewed using Alt + F<x>.[4]

After the first part of the installation, the system is partially or totally rebooted. In rare cases the system hangs when starting the PCMCIA system. The installation

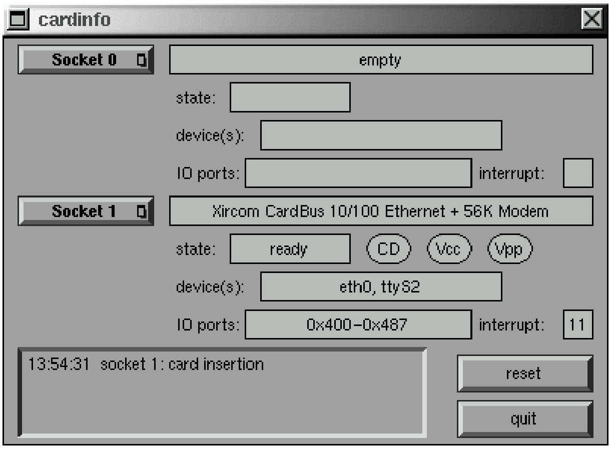
[3] Tip: **`rpm -ql pcmcia | grep man`** gives a list all package `pcmcia` manpages.

[4] In YaST2 you must use Ctrl + Alt + F<x> because YaST2 runs under the X Window System.

is far enough along that you can use the boot option **NOPCMCIA=yes** to boot Linux without PCMCIA support, at least in text mode. More information on this can be found in Section 7.1.5 page 196.

7.1.7 Other Help Programs

The program cardctl has already been mentioned several times. cardctl is an important tool for finding information about PCMCIA devices or taking specific actions. Details can be found in manpage for **cardctl** (**man cardctl**) or simply give the command **cardctl** to receive a list of command options.



cardinfo

Socket 0: empty
state:
device(s):
IO ports:
interrupt:

Socket 1: Xircom CardBus 10/100 Ethernet + 56K Modem
state: ready
device(s): eth0, ttyS2
IO ports: 0x400-0x487
interrupt: 11

13:54:31 socket 1: card insertion

Figure 7.1: PCMCIA – Cardinfo

This program has a graphical front-end, as well, which can be started with the command **cardinfo** (See Figure 7.1). Most options can be controlled with this although, unfortunately, you cannot change between “schemes” using this tool.

Friends of KDE can also use the program kardinfo. This tool is basically the same as **cardinfo**.

More help can be found in package `pcmcia` with **ifport**, **ifuser**, **probe** and **rcpcmcia** although they are not often needed in every day problems. To discover all that exists in package `pcmcia`, use the command **rpm -ql pcmcia**.

7.1.8 Compiling PCMCIA

If you update the kernel and do not find a compatible PCMCIA package on the SuSE FTP server, you will then have to compile the PCMCIA package yourself. It is important that the correct kernel is running when you are recompiling, since some information will need to be extracted from there. The package `pcmcia` should already be installed but should not be started; to be safe, run another **rcpcmcia stop**. Then install the PCMCIA source package using YaST and, following that, enter:

```
earth: # rpm -ba /usr/src/packages/SPECS/pcmcia.spec
```

That was it. Now even a new binary packet can be found in `/usr/src/packages/RPMS` which you can use for other systems, given that you implement the same kernel.

7.2 APM – Power Management

7.2.1 Fundamentals

Power management can be used, assuming that the necessary hardware and BIOS routines exist. Most notebooks and desktops include these – the diverse energy saving functions are usually only important for mobile computers. These functions are described below:

Standby – In this operating mode the display is turned off and, by some machines, the processor power is reduced.

Suspend (to memory) – In this mode all system information is written to RAM and the entire system is then suspended. Here the computer uses very little energy; so little, in fact, that the computer can run in this mode from between 12 hours to several days on one battery. The advantage of this is that within seconds you can resume working at the same point at which you left, without rebooting or restarting any programs. This is where using Linux becomes interesting – you never need to turn the computer off – other operating systems become unstable in time. With most modern notebooks you only need to close the lid, suspending the system, and simply open it again later to resume working.

Hibernation (Suspend to disk) – In this operating mode the computer can run right through the winter. That is, all information is saved to the hard disk and the system is turned off. Resuming operation takes about 30 - 90 seconds and the system is back to the same point where you left off. Some manufacturers offer interesting mixes of suspend and hibernation.

Battery Control – Very interesting.

Automatic Shutdown – Interesting for desktops. After a “shutdown” the computer is completely turned off.

hard disk power off – This function saves not only energy but, for those with a loud hard disk, also stress. Problems can occur, however, with, for example, an editor that automatically saves changes, causing the drive to turn back on, again and again.

Some of these functions are implemented by the BIOS alone. Standby and suspend can be invoked, on many notebooks, with a combination of keys or by simply closing the lid. The operating system has nothing do with these functions although with the correct kernel and packages installed these can also be invoked with a command. This is helpful if you like to perform certain actions every time the mode is invoked.

7.2.2 The Correct Kernel

For advanced APM functionality a kernel which supports this is needed. Several of the SuSE standard kernel includes this functionality. You can check to see whether these functions are supported with the command **`cat /proc/apm`**. If a line appears with diverse numbers everything is okay and the command **`shutdown -h`** should turn the computer off. Because some BIOS's do not correctly support the standards strange things can sometimes happen. Some of these problems can be solved by using a special kernel configurations. More information can be found in our Support Database at `http://sdb.suse.de/sdb/en/html` or on the "Linux Laptop Homepage" at `http://www.cs.utexas.edu/users/kharker/linux-laptop`.

7.2.3 The APM Daemon

The daemon apmd, found in package `apmd`, series `ap`, serves to watch the battery and can take specific actions when "standby" or "suspend" is started. This package is not always necessary but can often be of use. In order to start this at boot the variable **`START_APMD`** in `/etc/rc.config` should be set to `yes`. You can also start this with the command **`rcapmd start`**.

Several variables can be configured in `/etc/rc.config.d/apmd.rc.config`. Because this file contains comments about the settings of these variables we won't go into detail here.

- Although PCMCIA in SuSE Linux is compiled with APM support, you can still sometimes run into difficulties here. Some of the card drivers do not return properly from a suspension (e.g. `xirc2ps_cs`). Therefore, apmd can deactivate the PCMCIA system before the suspension and reactivate it afterward. To this end, the variable **`APMD_PCMCIA_EJECT_ON_SUSPEND`** is set to `yes`.
- If the clock time is no longer correct following a suspension, **`APMD_SET_CLOCK_ON_RESUME`** is set to `yes`.
- If the X Window Systemis causing problems, **`APMD_LEAVE_X_BEFORE_SUSPEND=yes`** helps once the system is up again.
- Likewise the sound modules can be unloaded during suspension. **`APMD_STOP_SOUND_BEFORE_SUSPEND=yes`** functions for this purpose. Caution: when unloading, sound applications may quit or else the modules were not unloaded. If you are editing audio data, save it first to be safe.

In addition, you can set the "spindown" behavior of the hard disk differently for the battery mode or even tell the computer to shutdown when the battery reaches a critically low level. Advanced users can add functionalities themselves in the file `/usr/sbin/apmd_proxy`.

7.2.4 More Commands

The package `apmd` has a few other useful programs. With **`apm`** you can check the actual battery capacity and put the system into standby (**`apm -S`**) or suspend (**`apm -s`** mode); see also manpage for **`apm`** (**`man apm`**).

The command **`apmsleep`** suspends the system for a certain amount of time; see also manpage for **`apmsleep`** (**`man apmsleep`**).

To observe a log file without having the hard disk continuously running use the command **`tailf`** instead of **`tail -f`**.

There are also tools for the X Window Systemas well as the command **`xapm`** which shows a graphic representation of the battery status. If you're using KDE or at least kpanel – the program kbatmon shows battery status and can suspend or put the system to sleep.

7.2.5 Pause for the Hard Disk

The hard disk can be turned off when not needed with the program hdparm. This program also has other useful hard disk functions. The drive can be placed in standby mode with the command option **`-y`** or completely turned off with **`-Y`**. The drive will automatically turn off after 30 seconds with the option **`hdparm -S 6`**. The number in this command is the number of 5 second intervals (in this case 6, so $6*5 = 30$). This function is turned off with a value of `0`. With larger numbers the interval is also increased, more information can be found in the manpages for this program.

You can set the standby behavior to be dependent on whether the computer is plugged in or in battery mode by configuring `/etc/rc.config.d/apmd.rc.config` but if you do so be sure to set the variable **`APMD_CHECK_TIME`** to `0`.

Because the hard disk is often needed the sleep function is often useless because many programs, for example, text editors, regularly write temporary data to the drive. Programs intended to help, such as **`tail -f`** <**`logfile`**> often cause this as well. Replacing this command with **`tailf`** <**`logfile`**> solves this problem.

7.3 IrDA – Infrared Data Association

IrDA (Infrared Data Association) is an industry standard for wireless communications that uses light in the infrared spectrum. Many laptops delivered today have an IrDA compatible sender/receiver that enables the communication with other devices, such as printers, modems, LANs, or even other laptops. The transfer rate varies from 2400 bps up to 4 Mbps.

Note

Support for this protocol in the Linux kernel is not yet complete. The project is still seen as "experimental" (beta status). Therefore this funtionality is not included in the standard kernel. This implies that the driver has not been fully tested yet and may not be as stable as other parts of the kernel.

Software

The package irda, series n supplies support for infrared ports as well as the IrDA Protocol itself. After installation of this package you can find the documentation under /usr/share/doc/packages/irda/README.

>From the package howto you can install the IR-HOWTO. After installation it can be found under /usr/share/doc/howto/en/IR-HOWTO.gz. For more information on the Linux IrDA-Project visit the URL http://irda.sourceforge.net/.

Configuration

In a normal installation, the variables **START_IRDA**, **IRDA_PORT** and **IRDA_IRQ** are set in the file /etc/rc.config . Normally the use of IrDA is deactivated; with the command

```
earth: # rcirda start
```

you can manually activate the port. You can deactivate it with the parameter stop. When activated the necessary kernel modules are loaded.

With IRDA_PORT (Default /dev/ttyS1) and IRDA_IRQ (Default 3) you can configure the IrDA port; they are set by the script /etc/irda/drivers when support for the infrared port is activated.

If you give START_IRDA a value of yes support for the infrared port will be activated at boot. Unfortunately IrDA uses noticeably more electricity due to the so-called "discovery" packets that are broadcast every few seconds, which look for other devices with IrDA communication ports with which to communicate.

Application

In order to print using the infrared port you need to send the print data to the device file /dev/irlpt0. This device file has the same attributes as the normal "wired" port /dev/lp0, the only difference being the wireless transmission through infrared light.

You can configure a printer to use this port in the same manner as a parallel port or serial port printer using YaST (see Section 3.6.1 page 78). When printing be sure that the printer is in sight of the computer and that the IrDA support is started.

If you want to communicate with other computers, mobile telephones or the like, address the device using the device file /dev/ircomm0. With the Siemens S25 mobile telephone, for instance, you can use the program wvdial to access the internet.

Troubleshooting

If devices do not respond when using the infrared port, check to see if the device is found by the computer using the command **irdadump** as 'root':

```
earth: # irdadump
```

In the case of a Canon BJC-80 printer in “sight” of the computer `earth`, you’ll receive the following output in regular intervals (see Output 7.3.1).

```
21:41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21:41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21:41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21:41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21:41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21:41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21:41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80 \
                        hint=8804 [ Printer IrCOMM ] (23)
21:41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* erde \
                        hint=0500 [ PnP Computer ] (21)
```

Output 7.3.1: IrDA: **irdadump**

If you receive no output or the device does not respond, check the configuration of the port. Are you using the correct port? Sometimes the infrared port can be found under the device file `/dev/ttyS2` or `/dev/ttyS3` or the device is not being assigned to Interrupt 3. These settings can be changed on nearly every laptop in the BIOS setup.

It is important to note that you can only communicate with devices using IrDA if the device you wish to communicate with supports the protocols `Printer` or `IrCOMM`. With the help of special programs (**irobex_palm3** or **irobex_receive**, please make note of the descriptions in `IR-HOWTO`) you can communicate with devices that use the `IROBEX` Protocol (e. g.3Com Palm Pilot). Which protocol your device supports can be taken from the output of **irdadump** in brackets after the device name. Support for the `IrLAN` Protocol is still a “Work in progress” and will be included in future versions of Linux.

You can check to see if the IrDA port is actually emitting light with a normal video camera. In contrast to humans, most video cameras can “see” infrared light.

8 Printing

This chapter describes the essential procedures which occur "behind the scenes" when printing takes place.

8.1 Basics and Requirements for Printing

In Linux, data is normally sent to a printer via a "print queue". The "print spooler" is responsible for queuing the print jobs and sending them to the printer in the correct order, thus ensuring that they do not interfere with each other.

The data to be printed often requires conversion to a suitable format before the printer can handle it: graphics files must normally be first converted to a printable format, for example. The "print filter" is a piece of software which is responsible for this conversion.

8.1.1 Different Standard Printer Languages

ASCII text – Most printers can at least print out ASCII text directly. There are two general types of exception to this: those printers which first require the text to be converted to another of the printer languages listed below and the "GDI printers" which are designed only to work with Windows (TM) and, by default, are capable of printing neither ASCII text or data offered to them in any of the other standard printer languages.

PostScript – PostScript is the standard printer language in Unix/Linux. If no PostScript printer is available, a special filter program (Ghostscript) is used in Linux to convert PostScript data into one of the following standard printer languages.

PCL 5 or PCL 6 – These printer languages are common among laser printers.

PCL 3 – This printer language is used by most HP DeskJet printers and compatible models.

ESC/P, ESC/P2 and ESC/P Raster – These printer languages are used by most Epson printers and compatible models.

8.1.2 The Problem with GDI Printers

Many printers are sold as "Windows printers" or "GDI printers". (GDI stands for the Windows Graphical Device Interface: such printers are designed to work with only one operating system.) They are often difficult or impossible to set up to work with Linux: some of them are capable of using other standard printer languages and are thus usable, while others will only work at all with Windows (TM). You should consult the CDB, at `http://cdb.suse.de/`, or check with the hardware manufacturer if you are unsure.

With GDI printers, the manufacturer does without a standard protocol completely and controls the printer directly with control sequences of the specific model. However, there are printers on the market which can act both as GDI printers and also work with "proper" printer languages.

8.1.3 Hardware and Software Requirements

You should check the following points:

- You should check whether your printer is supported by SuSE Linux. You can do so by looking at the lists which can be found on the installed system and on the internet at:
  - `/usr/share/doc/packages/ghostscript/catalog.devices`
  - `http://cdb.suse.de`
  - `http://www.linuxprinting.org` → 'The Database' (`http://www.linuxprinting.org/database.html`)
  - `http://www.cs.wisc.edu/~ghost/printer.html`

> **Note**
>
> You will sometimes hear of printers which require the current version of "Aladdin Ghostscript". For licensing reasons, SuSE Linux includes the slightly different "GNU Ghostscript". In most cases, a GNU Ghostscript driver can be found which is suitable for any given printer.

- You should check that you can make the necessary physical and logical connection to the printer. See Section 8.2 on the facing page for details of this.
- You should note that a self-compiled kernel could lack the necessary support for printing: if in doubt, you should first install and boot from a standard SuSE kernel. (This can be done through YaST.)
- It is recommended that you install the '`SuSE Default System`' packages using YaST or the '`Standard-System`', using YaST2. The following packages are necessary along with the '`SuSE Minimal System`' for drives with little space to print ASCII and PostScript files:
  - package `a2ps`, series `ap`
  - package `aps`, series `ap`
  - package `gs_x11`, series `ap`
  - package `gs_fonto`, series `ap`

  - package gs_fonts, series ap
  - package gs_lib, series ap
  - package lprold, series n

  For the installation of additional packages, see Section 3.4 page 71.

If in doubt, you can orient yourself with the current version of the SDB article file:/usr/share/doc/sdb/de/html/jsmeix_print-einrichten.html.

Possible Problems

- If you have updated an older SuSE Linux, there may be inconsistencies in the printing system's packages and (configuration) files. In this case, it makes sense to reconfigure the entire printing system.

8.1.4 Defining a Suitable Printer Driver

In the lists mentioned in Section 8.1.3 on the facing page Ghostscript drivers for specific printer models are provided. If your supplier cannot give you any Linux-specific information on your printer model, the following tips may help you:

- Find out if your printer is compatible with a model which does run in Linux and then use the Ghostscript driver for this compatible model.

  "Compatible in Linux" means that your printer can print correctly with the same binary control sequences as the compatible model – i. e. the printer can "understand" the same printer language directly (rather than just being capable of emulating it in Windows through the use of driver software).

  The fact that two printers have similar names is not always a guarantee that they are compatible. This is because printers with similar names do not always understand the same printer language directly.

- The best way of finding out what printer language your printer understands is from the manufacturer. The printer language is often also specified in the technical data of the printer manual. The most common printer languages are listed in Section 8.1.1 page 205.

- Ghostscript includes a special uniprint driver with parameter files for various printers with the extension .upp. If a parameter file exists for your printer, it can be configured in this way.

8.2 Connecting the Printer Locally

8.2.1 Parallel Ports

Connecting a printer to a Linux system is usually done via a parallel port. A printer on a parallel port is addressed via the parport subsystem of the kernel.

The parallel ports are made known to the `parport` subsystem by loading additional architecture-specific drivers (see Section 6.4.3 page 179.). In this way, a number of devices in series can be used *simultaneously* via *one* parallel port. The numbering of device files for parallel port printers starts with `/dev/lp0`. To be able to print via the first parallel port, the modules `parport`, `parport\_pc` and `lp` must be loaded – this is normally performed automatically via the kmod (*Kernel Module Loader* - see Section 9.2 page 239), as soon as a print job is sent. If the parallel port is not found, you can remedy this by using kernel parameters.

Configuring the first parallel port

The first parallel port, `/dev/lp0`, is configured by an entry in `/etc/modules.conf` (File 8.2.1).

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=none,none
# If you have multiple parallel ports, specify them this way:
# options parport_pc io=0x378,0x278  irq=none,none
```

File contents 8.2.1: `/etc/modules.conf`: First parallel port

`io` stands for the first address of the parallel port. The entry **`irq=none`** none puts the parallel port into "polling mode" rather than interrupt driven mode. This is generally a better option and is slower only on very old computers.

In order for these settings to work, the following settings for the first parallel port must be made in the BIOS, or via the computer's firmware:

- IO-address `378` (hexadecimal)
- Interrupt `7` (not relevant in polling mode)
- Mode `Normal` or `SPP` (other modes do not always function)
- `DMA` is switched off (should be switched to `Normal` mode)

Testing the first parallel port

If the printer is able to print ASCII text, then, as user `'root'` (System administrator), you should be able to print out a page with the word `hello`, using the command

```
earth: # echo -en "hello\f"  >/dev/lp0
```

.

Configuring the second parallel port

A second parallel port, which can be addressed via the standard IO address 278 (hexadecimal) (e. g. which can be set by jumper on an ISA port card), can also be configured with an entry in `/etc/modules.conf` (File 8.2.2 on the next page).

After a reboot or restarting all services (see page 210), the second parallel port will be available.

```
alias parport_lowlevel parport_pc
# options parport_pc io=0x378 irq=none,none
# If you have multiple parallel ports, specify them this way:
options parport_pc io=0x378,0x278  irq=none,none
```

File contents 8.2.2: `/etc/modules.conf`: Two parallel ports

Testing the second parallel port

If the printer is able to print ASCII text, then, as user `'root'`, you should be able to print out a page with the word `hello`, using the command

```
earth: # echo -en "hello\f" > /dev/lp1
```

.

8.2.2 Special Slot Cards: ISA-PnP and PCI

If the IO address of an additional parallel port is not known beforehand, this must be found out.

ISA PnP cards

Sometimes fixed values for IO address, interrupt and mode can be set on such cards, either by jumper or – if you have DOS/Windows – via an included DOS/Windows program. If this is possible, you should set fixed values for IO address, interrupt and mode.

Otherwise the values for IO address, interrupt and mode are only entered on the ISA PnP card when Linux is started up. The values entered can either be seen in the Linux boot messages, or you can proceed as described in Chapter 6.2 page 174.

PCI cards

Which IO addresses and which interrupts can be considered for a PCI card can be seen from the following command (see Output 8.2.1):

```
earth: # /sbin/lspci -vv | less
```

```
...

00:0a.0 Parallel controller: ...
        ...
        Flags: stepping, medium devsel, IRQ 10
        I/O ports at b400
        I/O ports at b000
        I/O ports at a800
        I/O ports at a400
...
```

Output 8.2.1: Extract from **`lspci -v`** for a PCI port card

In each case two IO addresses, separated by 400 (hexadecimal), belong together with a parallel port. The configuration entry must appear as shown in File 8.2.3.

```
alias parport_lowlevel parport_pc
# options parport_pc io=0x378 irq=none,none
# If you have multiple parallel ports, specify them this way:
options parport_pc io=0x378,0xb400,0xa800 irq=none,none,none
```

File contents 8.2.3: `/etc/modules.conf`: PCI card with 2 parallel ports

After a reboot or restart of all services (as described below), all three parallel ports will become available.

Testing all parallel ports

If you have a normal (ASCII-capable) printer, you should, as the user `'root'`, have a page printed out with the word `hello`, using the commands

```
earth: # echo -en "hello\f" > /dev/lp0
earth: # echo -en "hello\f" > /dev/lp1
earth: # echo -en "hello\f" > /dev/lp2
```

Instead of rebooting, it is enough, as the user `'root'`, to update the list of kernel module dependencies and – if only the printer is connected to the parallel ports – stop the printer daemon, unload the kernel modules which concern the parallel ports, reload them again and restart the printer daemon:

```
earth: # /sbin/depmod -a
earth: # rclpd stop
earth: # rmmod lp
earth: # rmmod parport_probe
earth: # rmmod parport_pc
earth: # rmmod parport
earth: # modprobe parport
earth: # modprobe parport_pc
earth: # modprobe parport_probe
earth: # modprobe lp
earth: # rclpd start
```

8.2.3 USB Connections

If you intend to use a USB printer, you should check that USB support is enabled in the BIOS. In an Award BIOS, for example, in the menu '`PNP AND PCI SETUP`' (or similar), the entry '`USB IRQ`' (or similar) must be set to `Enabled` (or similar).

You can test if the USB printer reacts by entering, as the user root:

```
earth: # echo -en "hello\f" > /dev/usblp0
```

Providing that there is a printer on the first USB port, `/dev/usblp0`, that can print ASCII text, a page with the word `hello` should be printed.

It may sometimes be the case that the USB printer can no longer be addressed – e. g.if you remove the USB plug during printing. Then you will need to stop all processes which access /dev/usblp0, unload the kernel modules involving the USB printer and reload these again, by entering the following commands as user root:

```
earth: # fuser -k /dev/usblp0
earth: # rmmod printer
earth: # rmmod usb-uhci        # or:   rmmod usb-ohci
earth: # umount usbdevfs
earth: # rmmod usbcore
earth: # modprobe usbcore
earth: # mount usbdevfs
earth: # modprobe usb-uhci     # or:   modprobe usb-ohci
earth: # modprobe printer
```

Check beforehand, with **lsmod**, which USB modules are loaded (either usb-uhci or usb-ohci) and if further module dependencies exist; for example, the display

```
usbcore  ...  [printer usb-uhci]
```

means that the module usbcore is still needed by the modules printer and usb-uhci. In this case, therefore, the modules printer and usb-uhci must be removed before the module usbcore.

8.2.4 Serial Ports

To set up a serial printer, use **lprsetup**. This menu-driven configuration script is described in Section 8.6 page 217.

After ending **lprsetup** the lines shown in 8.2.4 in the file /etc/printcap must appear in addition to the normal entries.

```
        :br#9600:\
        :ty=ixon -imaxbel -ixany -ixoff -crtscts:\
```

File contents 8.2.4: /etc/printcap: Serial ports

With **br**, the baud rate of the serial port is specified. **ty** stands for the **stty** options. You may need to adjust both values, depending on the hardware you have.

If there are problems, you can use, instead of the standard spooler from the package lprold, the print spooler from the package plp as follows (both packages are in the series n):

1. Log in as user 'root'.
2. Stop the print spooler with:
   ```
   earth: # rclpd stop
   ```
3. Uninstall the package lprold, using YaST.
4. Finish YaST and restart it to install the package plp.
5. Start up the print spooler with:
   ```
   earth: # rclpd start
   ```

Additional information

The parameters of the serial port can be set with the program **`setserial`**; for this, see the manpage for **`setserial`** (**`man setserial`**).

If a matching **`setserial`** command is written in `/etc/init.d/boot.local`, this will be automatically carried out each time the computer is booted (see also `/etc/init.d/serial`).

In the manpage for **`stty`** (**`man stty`**) many options are explained.

8.3 lpd: the Print Manager and Tools

8.3.1 The Print Manager in the Background

The lpd `Line Printer Daemon` is launched by the `/etc/init.d/lpd` script and runs in the background, assuming the corresponding option **`START_LPD=yes`** has been set in `/etc/rc.config`.

lpd checks `/etc/printcap` to see which print queues are defined. Its job is to organize the execution of spooled jobs:

- it manages local print queues: it sends every job's data file through the appropriate filter—which is assigned by the entry in `/etc/printcap` and by an explicit setting in the control file—and then to the printer port;
- it takes care of the order of the jobs;
- it checks the `status` of queues and printers in the local spool directories;
- it sends jobs to an lpd on a remote machine;
- it accepts requests from remote machines for local queues or refuses them if authorization fails.

Authorization: only requests from hosts listed in `/etc/hosts.lpd` are executed. It is also possible to enter the name in `/etc/hosts.equiv`, but this has *far-reaching* consequences as far as access and security are concerned and, therefore, should be avoided.

Additional definitions of the queue can restrict access to certain users, or users with an account on the local machine.

8.3.2 Additional Tools and Control Tools for the User

If you are printing a file, for example via the `raw` queue, which is not available in printer-specific format, the printer will not know what to do with this data: large amounts of paper will be printed with meaningless characters. The same thing will happen if the print filter is not properly configured, and produces printer-specific data for an incompatible printer model.

If you want to remove such a print job from the queue you can use the following commands to process the print queues.

Note

You need to have `'root'`-permissions to be able to manipulate the print queues, i.e. if you want to delete other people's print jobs, for example, you must work as `'root'`.

lpq shows your own jobs in the queue: For example:

```
newbie@earth: > lpq -Pqueue
```

```
queue is ready and printing
Rank   Owner      Job  Files        Total Size
active newbie     676  Hello.txt    259420 bytes
1st    newbie     677  letter.dvi    11578 bytes
2nd    newbie     683  picture.gif   37464 bytes
```

Output 8.3.1: Example output of the command **`lpq`**.

lprm removes your own jobs from a queue. Thus entering this:

```
newbie@earth: > lprm -Pqueue 676
```

will produce the following output:

```
dfA676Aa05005 dequeued
cfA676Aa05005 dequeued
```

Output 8.3.2: Example output of the command **`lprm`**.

A print job belongs to the user who initiated it. This user, as well as the job number, are displayed with the **`lpq`** commmand. The same job number can appear in different queues, so you always need to give the name of the queue.

If no job number is given, the currently active job is removed from the relevant queue—if it is your own job.

`lprm -Pqueue` deletes the current job from the specified queue, as long as this belongs to the user who issued the **`lprm`** command. The printer will still print the data which is already in its buffer.

`lprm -Pqueue` deletes all print jobs of the user from the specified queue. If `'root'` enters this command, all print jobs are deleted from the specified queue.

lpc controls the print queues: these can be specified by name, or you can specify **`all`**. Most of the lpc commands listed below can only be executed by "root". The most important options are:

`status queue` Provides a status report. If the <queue> argument is missing then this functions like <all>, i. e. a status report for all queues is displayed.

`disable queue` Stops new jobs being moved to the queue.

enable queue Enables the queue to accept new jobs.

stop queue Stops jobs from the queue being printed (the job currently being printed is completed).

start queue Starts printing jobs again from the queue.

down queue Is the same as **disable** plus **stop**.

up queue Is the same as **enable** plus **start**.

abort queue Is the same as **down**, except that a job currently being printed is interrupted immediately. The jobs remain, however, and printing can be continued after the queue is re-started (**up**).

You can pass these commands on to **lpc** directly in the command line (e. g. **lpc status**). Or you can run **lpc** without parameters: then the dialog mode with its own prompt, **lpc>** is started, awaiting the entry of **lpc**-commands. You can end this dialog with **quit** or **exit**.

> Note
>
> Print jobs remain in the queues even when you shut down the computer during printing and then start Linux again. A print job which has gone wrong should be removed from the queue with the commands shown above.

8.4 lpr: Sending off Print Jobs

The lpr is the daily “user interface” for existing print queues. The following procedure takes place:

1. Either the user himself causes a new print job with **lpr** or an application uses the **lpr** command, when the user chooses its ‘Print’ menu item.
2. **lpr** saves the data to be printed to the print queue. From there the print spooler forwards this to the print filter belonging to the print queue.
3. The printer filter converts the data to be printed to a format which the printer can print directly and then sends this to the printer.
4. After the entire print job has been sent to the printer it is removed from the print queue.

Normally a job is started using:

```
newbie@earth: > lpr text_file
```

or:

```
newbie@earth: > lpr [-Pqueue] text_file
```

If you omit the option **-P** (for “Printer”, the queue’s name), the default is taken from the PRINTER environment variable. If this is not set, the default name **lp** is used. The same applies for the commands **lpq**, **lprm** and **lpc** (see below).

Various print queues are needed for the following reasons:

- If a number of different printers are available, these different printers are addressed via different print queues, such as with

  ```
  newbie@earth: > lpr -Pprinter2 file
  ```

  for a second printer via the print queue `printer2`.

- Each print queue uses its own print filter. So for example you might use a standard print queue for rapid black and white printing and another print queue (for example with the name `color`) with a slower print filter which provides higher quality color printing: you would access this by

  ```
  newbie@earth: > lpr -Pcolor file
  ```

If you have configured the printer with YaST1, YaST2 or **`lprsetup`**, the following queues are always created by default:

`lp` – If you print a file via this standard queue, the apsfilter is used as the print filter. The filter automatically determines what type of data the file contains and conversion to the printer-specific format is carried out accordingly.

`ascii` – If you print a file via the `ascii` queue, with

```
newbie@earth: > lpr -Pascii file
```

then the apsfilter is used as the print filter, but this does not automatically define what type of data is contained in the file, rather the command line specifies that the file contains ASCII text, and conversion to printer-specific format is carried out accordingly.

`raw` – If you print a file via the `raw` queue, with

```
newbie@earth: > lpr -Praw file
```

then the apsfilter is used as the print filter, but this does *not* carry out a conversion to printer-specific format – rather the file is sent “raw” to the printer. So in this case the file must already contain data in printer-specific format.

In all these cases the same printer is always used, but this functions differently, depending on the print queue involved.

Tip

For especially large print jobs, it may be useful, instead of making a copy of the data file, to make a symbolic from the file to the spool directory. This can be done using the lpr option **`-s`**. Of course, you should then remove write permissions to this file until the print job has been completed!

8.5 Configuring a Printer with YaST2

Initially YaST2 checks to see if a configuration already exists – i. e.YaST2 considers a configuration already at hand. If no apsfilter entry has been found in `/etc/printcap`, an existing `/etc/printcap` will be renamed to `/etc/printcap.YaST2save` and a new configuration will be added. In order to force a new configuration, the /etc/printcap file should be moved with:

```
mv /etc/printcap /etc/printcap.save
```

Required Information

In Linux, the required data will be created by the print job processing itself: Port, queue name, printer model-specific settings (Ghostscript drivers) and hardware-independent settings (layout and country-specific ASCII text coding).

About the Interface

`/dev/lp0`	is the first parallel interface
`/dev/usblp0`	is the interface for a USB printer

For manual configuration, the interface must be specified. For this, a test should be run to see if the printer actually responds via the specified interface, because, if the printer is not responding, continuing configuration, of course, does not make much sense. It will function best if the printer is connected directly to the first parallel port and if the following settings for the parallel interfaces have been entered in the BIOS:

- IO address 378 (hexadecimal)
- Interrupt is irrelevant (by default, no interrupt is used)
- Mode `normal`, `SPP` or '`output-only`' (other modes do not always work)
- DMA is disabled (should be automatically deactivated in the above mentioned modes)

About the Queue Names

Since you will frequently need to enter the queue name, you should only give the queues short names consisting of lower-case letters and possibly numbers. By default, an additional `ascii` queue is added to each queue automatically. More queues may be automatically added which may or may not be needed (see **`lpc status`**), along with the `ascii` queue. With **`lpc down queue`**, unnecessary queues can be deactivated.

Queue names in a default configuration with YaST2

1. lp – print a file over the default queue using

   ```
   lpr filename
   ```

   which is how the apsfilter automatically defines which data type the file contains and the conversion will proceed accordingly to the printer-specific format. There should always be a queue with the name `lp` since this is the traditional name of the default queue.

2. lp-ascii – print a file via the ascii queue using

   ```
   lpr -Plp-ascii filename
   ```

   which states for the apsfilter that the file contains ASCII text and the conversion to the printer-specific format will take place accordingly. The `ascii` queue is necessary to force the output of ASCII text for ASCII text files that are not automatically recognized as such by the default queue.

3. lp-raw – print a file over the `raw` queue using

   ```
   lpr -Plp-raw filename
   ```

   and the apsfilter will not convert into the printer specific format, but rather, the file will be sent to the printer in raw form. Thus, in this case, the file has to already contain data in the printer-specific format. The `raw` queue is necessary whenever an application has already generated printer-specific data.

8.6 SETUP (`lprsetup`): Configuring the Print System

The package `aps` installs the configuration program /var/lib/apsfilter/SETUP. This is normally started with the command lprsetup which is symbolically linked to it. It provides the following services:

- A list of all apsfilter printer configurations
- Allows insertion and deletion of apsfilter printer configurations

8.6.1 Hints on Configuring with lprsetup

`lprsetup` is menu-driven and can be operated with the keyboard. Moving between the answer fields '`OK`' and '`Cancel`' is done in all menus by pressing the Tab key; use the ↓ and ↑ to move through selection lists.

1. Start **`lprsetup`** as the user '`root`' on the text console:

   ```
   earth: # lprsetup
   ```

   Press ↵ and you will reach the main menu:

   '`EXIT`' *Exit apsfilter setup* – end **`lprsetup`**.

   '`LISTING`' *List all apsfilter entries* – list all existing queues.

   '`ENTRY`' *Add/Overwrite/Delete an apsfilter entry* – To add a new printer the menu item '`ENTRY`' is used. This leads to another menu, '`Choose your printer definition`', in which you define your printer.

   '`DELETE`' *Fast delete an apsfilter entry* – To delete an existing apsfilter-printer, select '`DELETE`', and you will receive a list of current apsfilter-printers to choose from.

2. Then select '`ENTRY`'. You will see the following:

RETURN	Back to previous menu
DEVICE	Change printer interface

3. Select ‘`DEVICE`’ (Port).

   For printers on a parallel port or a USB port, choose ‘`PARALLEL`’ and for ones on a serial port, choose ‘`SERIAL`’. Parallel ports: `/dev/lp0`, `/dev/lp1`, `/dev/lp2`; USB printer ports: `/dev/usblp0`, `/dev/usblp1`, `/dev/usblp2`; serial ports: `/dev/ttyS0`, `/dev/ttyS1`, `/dev/ttyS2`.

   Confirm your entry with ‘`OK`’.

4. Select ‘`PRINTER`’ and choose the correct printer type from the following list:

POSTSCRIPT	For a real PostScript printer
HEWLETT-PACKARD	For an HP DeskJet printer
OTHER	Another non-PostSript printer
FREEDEF	Free definition of driver name

   Confirm this with ‘`COMMIT`’.

   Look through the following list for the matching Ghostscript driver or parameter file (recognized by the ending `.upp`) and then click on ‘`OK`’.

5. Enter the resolution. Only leave the defaults if you do not know the correct resolution. If you are unsure, try the resolutions 300, 360, 600, and 720. If you are using a `.upp` paramter file, it is important that the resolution you set here should match that given in the parameter file. Then click on ‘`OK`’.

6. Select ‘`PAPER`’ and go through the following list until you have found the correct paper size (for laser printers normally `a4` and for Inkjet printers normally `a4dj`). Then click on ‘`OK`’.

7. Select ‘`COLOR`’ and enter ‘`MONO`’ or ‘`COLOR`’ to make it easier for a later configuration to distinguish between black and white and color printing – for this, look at the next item and the following Section, “Configuring the apsfilter”.

8. If you have not yet configured a standard lp queue, **`lprsetup`** will always create the standard queues lp, `ascii`, `raw` and possibly also `lp-mono`; `lp-mono` is only created if you choose‘`COLOR`’ in the last item. If the standard queues were already set up you can, when configuring additional queues, enter your own queue names under ‘`SPECIAL`’ and ‘`NAME`’. This should only consist of letters and numbers, whereby short names with lower case letters are preferable, such as `color` for a color print queue.

9. *Important:* With ‘`ADD`’ the new printer is finally set up. A list is displayed of which print queues were created on the system with the apsfilter configuration.

   This means there are always several queues created, each enabling a different printer output. Make a note from this list of which print queues were created (normally only long lines are given – in the above example, you should note `color-ascii`, `color` and `color-raw`. Then choose ‘`OK`’.

10. Now the configuration files for the apsfilter are shown. Any changes should not be made in the global configuration file `/etc/apsfilterrc`, but only

```
### cdjcolor a4dj mono 300 color ###
color-ascii|lp10|cdjcolor-a4dj-ascii-mono-300-color
      |cdjcolor a4dj ascii mono 300
color|lp11|cdjcolor-a4dj-auto-mono-300-color
      |cdjcolor a4dj auto mono 300
color-raw|lp12|cdjcolor-a4dj-raw-color|cdjcolor a4dj raw
```

Output 8.6.1: List of print queues

in the configuration file allocated to the print queues just created. The latter configuration file has the form `/etc/apsfilterrc.` <`gs_driver`>, where <`gs_driver`> is the Ghostscript driver chosen in item 4 4.

If the same Ghostscript driver is chosen for a number of configurations you can still use the same configuration file, since different queues can be distinguished within such a configuration file. For more on this, see the following Section “Configuring the apsfilter”. Press ↩ to continue.

11. Now choose ‘`RETURN`’ and then ‘`EXIT`’.

8.6.2 Testing the printer

If the standard queues are created, you should now be able to print the file `/etc/fstab` with the command

```
newbie@earth: > lpr /etc/fstab
```

If the chosen Ghostscript driver allows color printing, you can test this with:

```
newbie@earth: > lpr /usr/share/ghostscript/5.50/examples/colorcir.ps
```

8.6.3 Special cases

With the command **`lpr file`** the type of file (e.g. ASCII or PostScript) is automatically defined. If the ASCII file type is to be forced, you can do this with

```
newbie@earth: > lpr -Pascii file
```

(e. g.to print a PostScript source text as such, as this will otherwise be interpreted as a PostScript file).

For files which already exist in printer-specific format, you should use:

```
newbie@earth: > lpr -Praw file
```

With

```
newbie@earth: > lpr -Plp-mono file
```

black and white print is forced. To enable this you must first set the relevant Ghostscript parameters in an `apsfilterrc` file; refer to Section 8.7.2 page 221.

8.7 The apsfilter

8.7.1 How the apsfilter Works

When configuring the printer with YaST1 or **lprsetup** the following parameters are saved in /etc/printcap:

- Ghostscript driver; e. g. hpdj, ljet4, stcany.upp.
- Paper format; e. g. a4, a4dj, letter, letterdj.
- Method; ascii or auto or raw.
- Color; if the method is not raw: mono or color
- Resolution; if the method is not raw: e. g. 300, 360, 600, 720.
- Queue name; if this is explicitly given with **lprsetup** under 'SPECIAL - Settings for your printer'

The actual print filter, the shell script /var/lib/apsfilter/apsfilter, is called up via a link in accordance with the **if** definition in /etc/printcap. The above parameters are passed on via this link to the print filter.

```
color-ascii|lp4|hpdj-a4dj-ascii-mono-300-color\
    |hpdj a4dj ascii mono 300:\
    :lp=/dev/lp0:\
    :sd=/var/spool/lpd/hpdj-a4dj-ascii-mono-300-color:\
    :lf=/var/spool/lpd/hpdj-a4dj-ascii-mono-300-color/log:\
    :af=/var/spool/lpd/hpdj-a4dj-ascii-mono-300-color/acct:\
    :if=/var/lib/apsfilter/bin/hpdj-a4dj-ascii-mono-300-color:\
    :la@:mx#0:\
    :tr=:cl:sh:
```

File contents 8.7.1: queue for color in /etc/printcap

Example: If an entry exists in /etc/printcap something like what is shown in File 8.7.1, then the following parameters are passed on to the apsfilter:

Ghostscript driver	:	**PRINTER="hpdj"**
Paper format	:	**PAPERSIZE="a4dj"**
Method	:	**METHOD="ascii"**
Color	:	**COLOR="mono"**
Resolution	:	**DPI="300"**
Queue name	:	**QUEUE="color"**

Depending on how these variables are set, apsfilter will behave differently:

METHOD="raw" – The data is sent to the printer device without being converted (e. g. /dev/lp0).

METHOD="ascii" – It is assumed that the data is ASCII text and that this text will be converted with a2ps to the PostScript format; see the manpage for **a2ps** (**man a2ps**).

The data in PostScript format is then converted by Ghostscript (**gs**) to the printer-specific format and sent to a printer device. The relevant Ghostscript driver and the resolution are defined in the above variables; further Ghostscript parameters can be configured with the apsfilter .

METHOD="auto" – apsfilter automatically tries to define what type of file the data is. The required conversion program is used to transform the data to the PostScript format. The data in PostScript format is then converted by Ghostscript to the printer specific format and sent to the printer device.

8.7.2 Configuring the apsfilter

Through special variables, the behavior of the apsfilters can be individually adjusted. These variables are entered in one of the following configuraion files:

- /etc/apsfilterrc
- /etc/apsfilterrc.<gs_driver>

where <gs_driver> is replaced by the actual name of the Ghostscript driver. These configuration files are read in by the apsfilter in the given sequence, i. e. this driver specific file /etc/apsfilterrc. <gs_driver> overwrites the defaults in the global file /etc/apsfilterrc.

In /etc/apsfilterrc only global defaults for all print queues should be entered. Special settings should be written in the matching /etc/apsfilterrc. <gs_driver>.

If you configure the printer with YaST2 the parameters are in part saved as above in /etc/printcap, but especially in /etc/gs.upp/y2prn_<queue>.upp. Here <queue> stands for the standard queue lp, or else the name which was assigned with an additional queue (e. g. color). The actual print filter /var/lib/apsfilter/apsfilter here is also called up via a link in accordance with the if entry in /etc/printcap. The above parameters are passed on to the print filter partly via this link and partly via /etc/gs.upp/y2prn_<queue>.upp. In this file the Ghostscript parameters are stored which were set by YaST2 during the printer configuration. Only very special Ghostscript parameter values which can not be adjusted by YaST2 can be entered here. The variables which define the behavior of the apsfilter must be entered in one of the following configuration files:

- /etc/apsfilterrc
- /etc/apsfilterrc.y2prn_<queue>.upp

These configuration files are read in by the apsfilter in the sequence specified, i. e. the settings in /etc/apsfilterrc.y2prn_<queue>.upp overwrites the defaults in /etc/apsfilterrc.

In /etc/apsfilterrc there are many helpful commentaries on the individual variables. Here is an overview of the most important ones:

FEATURE allows you to define how ASCII texts are printed:

The options:

FEATURE=1 prints one page of ASCII text per sheet of paper, with header

FEATURE=2 prints two pages of ASCII text per sheet of paper, with header

FEATURE=1n prints one page of ASCII text per sheet of paper, without header

FEATURE=2n prints two pages of ASCII text per sheet of paper, without header

FEATURE=1l prints in landscape format, with header

FEATURE=1ln prints in landscape format, without header

GS_FEATURES enables special Ghostscript parameters to be given. The Ghostscript parameters are very varied, depending on the Ghostscript driver used. Information on specific parameter settings can be found in /usr/share/ghostscript/5.50/doc/Devices.htm and /usr/share/ghostscript/5.50/doc/hpdj/gs-hpdj.txt; refer also to Section 8.9 page 228.

Continuing the above example of the file 8.7.1 page 220:

So that the printer really can always print in black and white in the case of **$COLOR="mono"**, the section from file 8.7.2 should be written into /etc/apsfilterrc.hpdj, for example.

```
if [ "$COLOR" = "mono" ]; then
  GS_FEATURES="-sModel=unspec -sColorMode=mono"
else
  GS_FEATURES="-sModel=unspec -sColorMode=CMYK"
fi
```

File contents 8.7.2: /etc/apsfilterrc.hpdj: Settings for black and white

By means of the file file:/usr/share/doc/packages/ghostscript/doc/hpdj/gs-hpdj.txt, you can check if the settings are correct or even possible for your printer model.

Brief information on this:

- For **-sModel=**, you can choose between 500, 500C, 510, 520, 540, 550C, 560C, 850C, 855C and unspec.
- For **-sColorMode=**, you can choose between mono, CMY, CMY+K and CMYK.

For explanations on the various color models (see file:/usr/share/doc/packages/ghostscript/doc/hpdj/gs-hpdj.txt):

mono The printer only has a black cartridge, e.g. the models 500, 510 and 520.

CMY The printer can print either with the black cartridge or with the color cartridge. The cartridges usually have to be changed manually, e.g. the models 500C and 540.

CMY+K The printer has both a black and a color cartridge, but black and color cannot be mixed. So for color printing, black is mixed from the colors, e.g. the models 550C and 560C.

CMYK The printer has both a black and a color cartridge, and black and color can be mixed, e.g. the models 850C, 855C and most other modern deskjets (unspec).

PRELOADS allows a file to be defined which is processed by the apsfilter before the actual data is printed. A number of examples of this:

- Adjusting the brightness when printing on color inkjet printers: See `file:/usr/share/doc/sdb/de/html/jsmeix_print-gammakorrektur.html`
- Adjusting page margins when printing PostScript files: See `file:/usr/share/doc/sdb/sdb/de/html/ke_print-margins.html`
- Switching a PostScript printer to duplex printing or choosing a different paper tray with a PostScript printer: See `file:/usr/share/doc/sdb/sdb/de/html/ke_print-duplex.html` and `file:/usr/share/doc/sdb/sdb/de/html/ke_tray.html`

ENCODING allows ASCII text and HTML pages from Netscape to be printed, in accordance with ISO-8859-1 or ISO-8859-2 – see also the comments in `/etc/apsfilterrc` and `file:/usr/share/doc/sdb/de/html/jsmeix_print-iso88592.html`.

8.7.3 Finding errors with apsfilter

A protocol recording what passes through the apsfilter when a print job is processed can be of help in looking for errors.

Procedure:

1. Log in as user '`root`'.
2. In the file `/var/lib/apsfilter/apsfilter` remove the commentary sign '`#`' in front of **`set -x`**.
3. From now on the activities of the print filter are recorded in matching log files which are specified in `/etc/printcap`. For **`lf`** the relevant log file of the specific print queues is given e. g.
   `/var/spool/lpd/hpdj-a4-auto-color-300/log`.

   The log file of the last printout in each case is appended to the existing log file, which can lead to these files becoming very large. You can empty these log files (without deleting them) using a command such as:

```
earth: # cat /dev/null > /var/spool/lpd/hpdj-a4-ascii-mono-
300/log
```

or

```
earth: # cat /dev/null > /var/spool/lpd/hpdj-a4-auto-color-
300/log,
```

but not deleted.

4. After a failed attempt to print, error messages in the log files can help you to find the cause of the printing problem. The log files in the above example can be browsed through using `less`, for example.

8.8 Printing in the TCP/IP Network

What's it all about?

If you

- use a printer in a TCP/IP network that is attached to another computer
- would like to use a printer via a TCP/IP network that is connected to a "printserver unit"
- wish to use a printer with a TCP/IP network connection

In order to accomplish, this you need to filter the print jobs using apsfilter.

Preliminary Remarks on Concepts

A computer in a TCP/IP network to which a printer is connected will be called a *print server*, in contrast to a *printserver unit*, which is a small device which connects a printer to a TCP/IP network. A printer that itself has a TCP/IP network connection will be referred to as a *network printer* and the computer that starts the print job as the *client*.

8.8.1 Overview: Specific Cases

The difference is in the filtering of the files to be printed. At some point in the chain, the files need to be converted to a format which the printer can print – in the print-language of the printer itself (PostScript, HP PCL, ESC/P). This is done by apsfilter. In the case of network printers and printserver units apsfilter runs on the client, which sends the information to the network printers or print server unit. In the case of a print server apsfilter may be run on the client or on the print server itself. There exist, then, several possibilities:

Network printers with filtering at the client:

1. Client: convert files to printable files (`-apsfilter->`)
2. Client: send the printable files to the network printer (`-remote->`)

Printserver unit with filtering at the client:

1. Client: convert files to printable files (`-apsfilter->`)
2. Client: send the printable files to the printserver unit (`-remote->`)
3. Printserver unit: send the printable files to the printer

A printer connected to a print server with filtering at the client:

1. Client: convert files to printable files (`-apsfilter->`)
2. Client: send the printable files to the print server (`-remote->`)
3. Print server: send the printable files to the printer

A printer connected to a print server with filtering at the print server:

1. Client: send the files to the print server (`-remote->`)
2. Print server: convert files to printable files (`-apsfilter->`)
3. Print server: send the printable files to the printer

- A printer forwarding queue is needed at the place where `-remote->` is. The information is not filtered here, only forwarded.
- At the place where `-apsfilter->` is, a printer forwarding queue is needed which filter the files. If the filtering is done at the client this called a “prefilter” for the “printer forwarding queue”. If the filtering is done at the print server, this is a normal printer forwarding queue such as `lp` or `ascii`.

Requirements:

The printer must be supported by SuSE Linux because the prefilter creates printable files in the same manner as that for a local printer. Refer also to Section 8.2 page 207.

There must be a TCP/IP connection to the network printer and it must be functioning properly.

8.8.2 Configuring a Printer Forwarding Queue on the Client

Log in as system administrator ‘`root`’ on a text console. If you have a graphical login switch to a text console, using `Ctrl` + `Alt` + `F2`.

1. Enter the command **`/var/lib/apsfilter/SETUP`** or **`lprsetup`**.
2. Press ⟵, to enter the menu ‘`APSFILTER SETUP`’.
3. Choose first ‘`ENTRY`’, then ‘`DEVICE`’ and then ‘`REMOTE`’.
4. Enter the IP address of the network printer, the printserver unit, or the print server. If your network has name resolution (DNS) you may alternatively enter the name of the print server.

5. Enter the description of the forwarding queue on the network printer, the printserver unit, or the print server. In the case of network printer or the printserver units you can find the possible descriptions in the documentation of the device. In the case of a print server is either `lp` or `ascii`, depending upon which is in use on the print server.
6. Choose '`ADD`'.
7. Press ⏎ .
8. Choose first '`RETURN`' then '`EXIT`'.

Now `remote` is configured on the client. After restarting the printer daemon on the client with the command

```
earth: # rclpd stop
earth: # rclpd start
```

the queue `remote` is ready for use.

With the command

```
earth: # lpr -Premote <file>
```

the <file> is sent without being filtered on the network printer or printserver unit. If the printer is connected to a print server the file `/etc/hosts.lpd` must include the IP address of the client. If name resolution functions you may also enter the name of the client here.

After restarting the printer daemon on the print server it will accept print jobs from the given client. If the printer is connected to the print server and the filtering is done on the print server you are now done with the configuration.

8.8.3 Configuring a Prefilter on a Client

1. Start **`lprsetup`**.
2. Press ⏎ .
3. First choose '`ENTRY`', then '`DEVICE`' and then '`PREFILTER`'.
4. Choose the configured queue `remote`.
5. Choose '`PRINTER`'.
6. Select the relevant print format from the list '`POSTSCRIPT`', '`HEWLETT-PACKARD`', '`OTHER`' or '`FREEDEF`'.
7. Choose '`COMMIT`'.
8. Look through the following list until you've found the correct Ghostscript driver or parameter file and then choose '`OK`'.
9. Enter the correct resolution and then choose '`OK`'.
10. First press '`RETURN`' then '`PAPER`'.
11. Enter the correct paper size and then choose '`OK`'.

12. Choose ‘`COLOR`’.

13. Choose either ‘`MONO`’ or ‘`COLOR`’ and then press ‘`OK`’.

14. Choose ‘`ADD`’. You will receive output telling you which queue will be used on your system with the apsfilter configuration. Make a note of this then choose ‘`OK`’.

15. Press ⏎.

16. Press first ‘`RETURN`’ then ‘`EXIT`’.

After restarting the printer daemon on the client with the command

```
earth: # rclpd stop
earth: # rclpd start
```

your new prefilter queue on the client is ready for use.

With the command

```
earth: # lpr -Pprefilter-queue <file>
```

the <file> is sent to the `prefilter-queue` given. Once there it is converted by apsfilter into printable files and, through the `remote`-queue is sent to the network printer, printserver unit or print server.

8.8.4 Problems with Configuration

Checking the prefilter configuration

Connect the printer to the first parallel port on the computer you wish to use and configure the printer, just for test purposes, locally on the computer.

To do this, proceed with Section 8.2 page 207.

This tests whether the problems lie on the printer or the network. If the printer functions you’ll know the correct Ghostscript driver and the other parameters for the configuration were correct. Now you can configure your printer as a network printer, in accordance with the above instructions.

The network printer or printserver unit doesn’t work properly

There are occasionally problems with the print spooler, e. g.when more than one printer is attached or when more than one file at a time needs to be printed. Since these are connected with the print spooler in the printserver unit or in the network printer, nothing much can be done. It is therefore generally easier to connect the printer to a print server.

The following remedy may help, however: a single computer serves as a print server for the printserver unit or the network printer. All clients send print jobs to this print server, and only this print server takes on the spooling of print jobs and sends these on to the printserver unit or the network printer. Filtering can also take place here on the client or on the print server. Details of this can be found at: `file:/usr/share/doc/sdb/de/html/jsmeix_print-netzwerk.html`.

Background information:

The “Printer forwarding queue” and the “Prefilter-queue” cannot be contained in the same queue because the print daemon lpd (from package `lprold`) ignores distant queue filter information given in `/etc/printcap` – therefore a prefilter needs its own queue.

8.9 Some Words on Ghostscript

If you don’t own a PostScript printer, Ghostscript is the most popular choice for a filter. Ghostscript accepts PostScript files and contains many “printer drivers” for conversion purposes. Ghostscript is much more, though, than just a collection of useful printer drivers – Ghostscript can also prepare PostScript files for screen output (for gv) or convert them to PDF format.

Ghostscript – described in `file:/usr/share/doc/packages/ghostscript/doc/Use.htm` – is an extensive tool with many command line options. If you invoke Ghostscript from the command line, it presents you with its own input prompt, **`GS>`**. You can leave this by entering `quit`.

The help command

```
newbie@earth: > gs -h | less
```

lists the most important options, the version number of the program as well as the *current list of supported devices;* the `uniprint` details also appear here – the parameter files for `uniprint` are listed in `file:/usr/share/doc/packages/ghostscript/catalog.devices`.

8.9.1 Examples of Working with Ghostscript

In `/usr/share/ghostscript/5.50/examples` – the Ghostscript version 5.50 is included in SuSE Linux– here you will find a number of PostScript files.

`/usr/share/ghostscript/5.50/examples/colorcir.ps` is suitable for testing purposes. In X, the graphical desktop, you can display a PostScript file on the screen with the command **`gs`**:

```
newbie@earth: > gs /usr/share/ghostscript/5.50/examples/colorcir.ps
```

To close this, press Ctrl + c in the terminal window from which you started **`gs`**.

Converting a PostScript file to the printer-specific format for a PCL-5 or PCL-6 printer can be performed, for example, with the command

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sOutputFile=/tmp/out.prn \ -
sDEVICE=ljet4 -r300x300 \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
```

where the command should be written in a single line, and the backslash (‘`\`’) must be suppressed. Furthermore it is assumed that the file `/tmp/out.prn` does not yet exist.

Converting a PostScript file to the printer-specific format for an HP DeskJet (color printer with PCL 3) can be done with one of the following commands:

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sOutputFile=/tmp/out.prn \
-sDEVICE=hpdj -r300x300 \
-sModel=500 -sColorMode=mono -dCompressionMethod=0 \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sOutputFile=/tmp/out.prn \
-sDEVICE=cdjmono -r300x300 \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sOutputFile=/tmp/out.prn \
-sDEVICE=cdj500 -r300x300 \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sOutputFile=/tmp/out.prn \
-sDEVICE=cdj550 -r300x300 \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
```

(Every command is to be entered on each line without '\'.)

The conversion of a PostScript file to the printer-specific format for an ESC/P2 or ESC/P or ESC/P dot matrix printer takes place using one of the following commands:

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sOutputFile=/tmp/out.prn \
@stcany.upp \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sOutputFile=/tmp/out.prn \
-sDEVICE=stcolor -r360x360 \
-dBitsPerPixel=1 -sDithering=gsmono -dnoWeave -
sOutputCode=plain \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
```

Here you can see the difference in the command when using a parameter file for the `uniprint` driver and when using one of the other Ghostscript drivers. Since all driver-specific parameters are given in the uniprint parameter file, no further driver-specific parameters need to be given, in contrast to the other Ghostscript drivers.

After each of the above commands, the printer-specific data is now located in `/tmp/out.prn`, which can now be sent by '`root`' directly to the printer with the following command

```
newbie@earth: > cat /tmp/out.prn >/dev/lp0
```

assuming that the printer is connected to the first parallel port, `/dev/lp0`.

If an ASCII text file is to be printed with Ghostscript, this should first be converted to the Ghostscript format with

```
newbie@earth: > a2ps -1 --medium=A4dj --output=/tmp/out.ps textfile
```

and then, with

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sOutputFile=/tmp/out.prn \
<driver-specific Ghostscript parameter> \
/tmp/out.ps quit.ps
```

converted to the printer-specific format, which is then, as '`root`', sent to the printer with

```
newbie@earth: > cat /tmp/out.prn > /dev/lp0
```

You can find all the latest information on the *Ghostscript Homepage* (versions, printer support, etc.). This is located at the URL:
`http://www.cs.wisc.edu/~ghost/`.

8.10 Examples of Your Own Print Filters

8.10.1 Preliminary Notes

The aim of this section is not to present an alternative to the apsfilter, but rather to explain the background to printing in Linux, using the example of a simple self-created print filter. This means that, in special cases, it is also possible to use an additional print queue with your own printer filter, if the apsfilter cannot be properly configured for this special case. The examples given here are kept as simple as possible to illustrate the essential steps. This is why things like error correction measures in the bash scripts were left out.

We are assuming here that the printer is connected to the first parallel port, `/dev/lp0`. If the printer is connected to a different port onm the computer, then you should use the device of this port, rather than `/dev/lp0` (see Section 8.2 page 207).

A print filter receives the data to be printed from the print spooler, via the standard input. The print filter must convert this data into the printer-specific format and then output this via the standard output. The print spooler ensures that everything that the print filter sends to the standard output arrives at the printer device, `/dev/lp0`. The kernel in turn forwards everything which arrives at the printer device onto the defined port (e.g. to the IO-address `0x378`). The hardware ensures that everything that is sent to the IO-address `0x378` is sent via the parallel connection cable to the printer. The printer interprets this data stream and prints accordingly.

The following commands can usually only be carried out as the user `'root'`, since normal users may not directly access the printer device unless the user `'root'`, with the command

```
earth: # chmod a=rw /dev/lp0
```

has given all users permission to access the printer device directly.

Commands can be given as follows:

```
earth: # cat ASCII-text_file >/dev/lp0
```

Here it should be clear that `ASCII-text_file` needs to be replaced by the name of an existing ASCII text file.

8.10.2 A Simple Example of the Basic Method of Working

Through the command

```
earth: # echo -en "hello\f" >/dev/lp0
```

no print spooler or print filter becomes active, since the printer device `/dev/lp0` is addressed directly. Through this only the ASCII characters `'h'`, `'e'`, `'l'`, `‘l’` and `‘o’` are sent directly to the printer. The character `‘\‘f’’` here stands for the FormFeed ASCII character, causing a line feed on the printer. As long as the printer can print ASCII characters directly, it will print the word `hello` and issue the page.

With

```
earth: # cat ASCII_text_file >/dev/lp0
```

no print spooler or print filter will become active, since the printer device `/dev/lp0` is directly addressed. The ASCII characters from the ASCII text file are sent directly to the printer.

In Linux, two ASCII text lines are separated only by a linefeed ASCII character (line break). In the various DOS versions and in Windows (hereafter referred to as DOS/Windows) two ASCII text lines are separated by a line feed ASCII character and a carriage return ASCII character.

If you send an ASCII test file directly to the printer with the above command, this will normally appear on the page as follows:

```
First Line
           Second Line
                       Third Line
                                   ...
```

because the printer will perform a line return, but not a carriage return, since no ASCII carriage return character was sent.

You can, however, adjust the printer so that it carries out both a line feed and a carriage return when it receives an ASCII line feed character.

For printers which understand the PCL 3 printing language (HP DeskJets and compatible models), the printer can be adjusted with the escape sequence `\033&k2G`, so that it performs a line feed and a carriage return when receiving an ASCII line feed character.

With

```
earth: # echo -en "\033&k2G" >/dev/lp0
```

the escape sequence is sent to the printer and then with

```
earth: # cat ASCII-Textfile >/dev/lp0
```

the ASCII text file is printed with the correct line wrapping. You may still need to enter

```
earth: # echo -en "\f" >/dev/lp0
```

to have the last page removed from the printer.

Usually umlauts will not be correctly printed, however, since they are coded differently in DOS/Windows than in Linux and printers are normally set by default for DOS/Windows.

With

```
earth: # cp ASCII-Textfile ASCII-Textfile.ibmpc
earth: # recode lat1..ibmpc ASCII-Textfile.ibmpc
```

the ASCII text file is first copied to ASCII-text_file.ibmpc and then `ASCII-text\_file.ibmpc` is newly coded to conform with DOS/Windows.

If you now enter

```
earth: # cat ASCII-Textfile.ibmpc >/dev/lp0
```

both line wraps as well as umlauts should be printed correctly. Since both line wraps and umlauts in the ASCII-text_file.ibmpc are now coded in accordance with DOS/Windows, a special escape sequence is no longer needed to adjust the appropriate line wrapping behavior for the printer.

So, with

```
earth: # cp ASCII-Textfile ASCII-Textfile.ibmpc
earth: # recode lat1..ibmpc ASCII-Textfile.ibmpc
earth: # cat ASCII-Textfile.ibmpc >/dev/lp0
```

any ASCII-text file can be correctly printed on any printer which can print ASCII text.

If this works, the next step is to create a print filter which automatically takes care of this conversion of the ASCII-text file into the printer-specific format.

Following the pattern of `/var/lib/apsfilter`, a subdirectory for your own print filter is created and you change to this subdirectory (as the user `'root'`):

```
earth: # mkdir /var/lib/myprinterfilter
earth: # cd /var/lib/myprinterfilter
```

Create a bash script (as an ASCII text file) with the name `asciifilter`, as shown in File 8.10.1.

```
#! /bin/bash

# make a temporary file
INPUT="$(mktemp /tmp/asciifilter.$$.XXXXXX)"

# First store everything from stdin in $INPUT
# to have the input as a regular file
cat > $INPUT

# Recode the INPUT
recode lat1..ibmpc $INPUT

# Add a FormFeed at the end of $INPUT
# to get the last page out of the printer
echo -en "\f" >> $INPUT

# Send $INPUT to stdout
cat $INPUT

# Remove the INPUT file
rm $INPUT
```

File contents 8.10.1: `/var/lib/myprinterfilter/asciifilter`

Make this bash script executable for all users, with

```
earth: # chmod a+x /var/lib/myprinterfilter/asciifilter
```

Create a new print queue with **`lprsetup`**. Select any Ghostscript driver which you would normally not use. Resolution and paper size are not relevant. Select a driver which you would not normally use ‘MONO’. But for ‘SPECIAL Settings for your printer’ enter, under ‘NAME Optional name for the queue’, af as the name, which should stand for ascii filter, provided you don’t already have a print queue with this name. The queues are then created by lprsetup: af-ascii, af and af-raw.

Below, only the af queue is used. The queues af-ascii and af-raw are actually irrelevant, and probably cannot be used, since the Ghostscript driver selected will not function for the printer connected.

In /etc/printcap you will now find an entry similar to the one shown in File 8.10.2 (abbreviated).

```
af...
        :if=/var/lib/apsfilter/bin/...-af:\
        :la@:mx#0:\
        :tr=:cl:sh:
```

File contents 8.10.2: /etc/printcap: Your own filter

Stop the print spooler with

```
earth: # rclpd stop
```

and just change the line in the above entry in /etc/printcap

```
        :if=/var/lib/apsfilter/bin/...-af:\
```

to

```
        :if=/var/lib/myprinterfilter/asciifilter:\
```

and then, after saving /etc/printcap, start the print spooler again with

```
earth: # /sbin/init.d/lpd start
```

and now, with

```
earth: # lpr -Paf ASCII-Textfile
```

every user should be able to print via the new af queue.

This new queue does exactly the same as the apsfilter, as long as you set USE_RECODE_NOT_A2PS="yes" in an apsfilterrc.

To make sure that data is not sent unintentionally to the printer via the irrelevant queues, af-ascii and af-raw, the lines for these queues should, in each case, be changed from

```
        :lp=/dev/lp0:\
```

to

```
        :lp=/dev/null:\
```

8.10.3 A More Complex Example

Concern

The existing driver is not a Ghostscript driver. In this case you eventually have to create your own printer driver; you have to distinguish between two variations:

1. There is no Ghostscript driver available for this printer (e. g.the printer is a "GDI"-printer).
2. A Ghostscript driver is available for this printer, but there's another non-Ghostscript driver available which creates better output (e. g.a driver specially optimized for this printer).

Such drivers are normally used as an addition to the Ghostscript driver by converting the Ghostscript output to the printer specific format.

As this is the most complex situtation, this example can be used to create a separate printer driver later on.

Requirements

It is assumed that a driver exists with the current version of SuSE Linux or that a driver can be downloaded from the internet. It's also assumed that you can handle unix source (e.g. `.zip`- or `.tar.gz`-archives and `.rpm`-packages); see also Section 11.3 page 285. After unpacking such an archive you normally find installation instructions in files like `README` or `INSTALL` or in a subdirectory named `doc`.

Background-Information

In the case of a `.tar.gz`-archive it's normally necessary to compile and install the driver. We assume, that the driver is being installed as

```
/usr/local/bin/printerdriver
```

here.

Normally the driver simply converts the Ghostscript output into a printer specific format (see Section 8.9 page 228).

Therefore Ghostscript needs to be called like this to create a 600 dpi resolution image:

```
-sDEVICE=pbmraw -r600x600
-sDEVICE=pnmraw -r600x600
```

Please have a look at the documentation provided with the driver to find out exactly which Ghostscript driver and which resolution is needed.

The output of Ghostscript is forwarded to the driver via a pipe:

```
| /usr/local/bin/printerdriver Parameter
```

Check the documentation of the driver to find out which parameters are necessary.

The complete command looks like this

```
gs -q -dNOPAUSE -dSAFER -sOutputFile=- \
     -sDEVICE=<device> -r<resolution> \
     - quit.ps \
     | /usr/local/bin/printerdriver <parameter>
```

The variables <device>, <resolution>, `/usr/local/bin/printerdriver` and <parameter> need to be replaced by the values described in the driver documentation.

For a printer that is supported by Ghostscript, the command line is simplified to

```
gs -q -dNOPAUSE -dSAFER -sOutputFile=- \
     -sDEVICE=<ghostscript-driver> -r<resolution> \
     <opt-parameter> \
     - quit.ps
```

where <ghostscript-driver>, <resolution> and <opt-parameter> need to be replaced by valid values (refer to Section 8.9 page 228).

As Ghostscript expects input in the postscript format, the data needs to be converted to postscript before Ghostscript is called.

Normally it will only do a conversion from ASCII-text to postscript, as most applications create either postscript (e.g. Netscape and StarOffice) or ASCII-text (e.g. editors). The format of the data can be identified by the **`file`** command.

ASCII-test can be converted to postscript with the **`a2ps`** command. To get details on the various options of this tool please refer to manpage for **`a2ps`** (**`man a2ps`**).

Installation and usage

Create a directory `/var/lib/myprinterfilter` and change to that directory.

Create a bash script called `printerfilter` (refer to 8.10.3 on the next page).

You can use this example as it is. All you need to change is the variable **`GSDEVICE`** and **`DRIVER`**. Change them to fit to your printer.

For a printer supported by Ghostscript enter:

```
DRIVER='cat -'
```

For a postscript printer, that requires no Ghostscript, enter:

```
GS='cat -'
DRIVER='cat -'
```

Change the variables **`A2PSLAYOUT`** and **`A2PSENCODING`** as you prefer; for additional information, refer to manpage for **`a2ps`** (**`man a2ps`**).

Make this bash script executable to everyone on the system and create a printer queue with **`lprsetup`**. Have a look at Section 8.10.2 page 230 for details on how to do that.

If the printerfilter doesn't work as expected, you can have a look at the log file which is created during a printjob and records all the output of the commands. Refer to Section 8.7.3 page 223 to read more about this.

```
#! /bin/bash

# For debugging see for stderr in /var/spool/.../log
#set -x

# Define the program calls for the filter pipes

# The a2ps call
A2PSLAYOUT="-1 --center-title= --borders=yes"
A2PSENCODING=’--encoding=ISO-8859-1’
A2PS="a2ps $A2PSLAYOUT $A2PSENCODING --output=-"

# The gs call
# in GSDEVICE the Ghostscript device and parameters have to be
# set
GSDEVICE=’-sDEVICE=Device -rResolution’
GS="gs -q -dNOPAUSE -dSAFER -sOutputFile=- $GSDEVICE - quit.ps"

# The call for an optional printerdriver
DRIVER=’/usr/local/bin/printerdriver Parameter’

# make a temporary file
INPUT="$(mktemp /tmp/printerfilter.$$.XXXXXX)"

# First store everything from stdin in $INPUT
# to have the input as a regular file
cat > $INPUT

# Determine the FILETYPE of the data in $INPUT
FILESTRING=$(file $INPUT | tr ’A-Z’ ’a-z’)
case "$FILESTRING" in
  *postscript*)                      FILETYPE=postscript ;;
  *ascii*|*text*|*english*|*script*) FILETYPE=ascii ;;
  *)                                 FILETYPE=unknown ;;
esac

# Do the filter pipe according to the FILETYPE
case "$FILETYPE" in
  postscript)
     cat $INPUT | eval $GS | eval $DRIVER ;;
  ascii)
     cat $INPUT | eval $A2PS | eval $GS | eval $DRIVER ;;
  unknown)
     echo "Unknown filetype: $FILESTRING" |
mailx -s "pinterfilter" root ;;
esac

# Remove the INPUT file
rm $INPUT
```

File contents 8.10.3: A Complex Printerfilter

9 The Kernel

The standard SuSE kernel, which is written to disk after installation (and which is found in a correctly installed system under /boot), is configured to support as wide a range of hardware and other kernel features as possible. For this reason, this kernel is not specifically tuned to your own hardware. If you have made the right selection during the installation or update, then this kernel will be specifically optimized for your processor. However, it is possible that this kernel may not be ideal for your purposes, most of which relate to network server and routing tasks. In addition, a small amount of RAM will be wasted as a small number of unnecessary drivers are loaded.

Several Makefiles are provided with the kernel to help automate the process. These Makefiles handle nearly all the details for you. The only thing you have to do by hand is to make selections which correspond to hardware settings and other kernel features.

Installation support is not provided for kernels that are not included with SuSE Linux; we will be pleased to help you, however, in the context of our Professional Services.

> Note
>
> The description below is based on kernel series 2.4x and 2.2.x. Many of the issues mentioned here will be valid for 2.0.x, but they might differ in the details!

9.1 Kernel Sources

To build a kernel, the following packages must be installed: the kernel sources, (package kernel-source), the C compiler (package gcc), the GNU binutils (package binutils) and the include files for the C compiler (package glibc-devel). They are located in series D (Development). It is highly recommended to install the C compiler in any case, since the C language is inseparable from UNIX operating systems.

The kernel sources are located in /usr/src/linux. If you plan to experiment with different kernel sources, you can unpack them in different directories under the directory /usr/src and create a symbolic link, /usr/src/linux , to the current kernel source. This is what YaST does automatically.

Since there is a lot of software that relies on the sources being in /usr/src/linux, you should maintain this directory as a symbolic link to your current kernel source to provide an error-free compilation of system programs which need to access the kernel sources.

9.2 Kernel Modules

Many drivers and features no longer have to be compiled directly into the kernel, but can be loaded at runtime via kernel modules. Which drivers are to be compiled into the kernel and which are loaded as runtime modules is defined in the kernel configuration.

Kernel modules are located at `/lib/modules/<version>` , where `<version>` is the actual kernel version.

The use of this feature decreases the size and RAM requirements of the kernel, which is quite desirable. For this reason, it is best to use this feature wherever it is reasonable. Basically, components which are not required for the system to boot up may be built as modules. This makes sure that the kernel doesn't get too big to be loadable by the BIOS or a bootloader. Drivers which are required for boot, such as `ext2`, the SCSI drivers on a SCSI-based system, and similar items should be compiled into the kernel. In contrast, items such as `isofs`, `msdos`, or `sound`, which are not needed for starting up your computer system, may be built as modules.

Handling Modules

The following commands are available for your use:

- **`insmod`**
  insmod loads the requested module after searching for it in a subdirectory of `/lib/modules/<version>`. It is preferrable, however, to use **`modprobe`** (see below) rather than **`insmod`**, which should no longer be necessary.

- **`rmmod`**
  Unloads the requested module. This is only possible if this module is no longer needed. It is not possible to unload the **`isofs`** module (the CD-ROM file system), for example, while a CD is still mounted.

- **`depmod`**
  Creates the file `modules.dep` in `/lib/modules/<version>` where dependencies of all of the modules are defined. This is necessary to ensure that all dependent modules are loaded with the selected ones. If **`START_KERNELD`** is set in `/etc/rc.config`, this file is created each time the system is started.

- **`modprobe`**
  Loads or unloads a given module while taking into account dependencies of this module. This command is extremely powerful and can be used for a lot of things (e. g., probing all modules of a given type until one is successfully loaded). In contrast to insmod, modprobe checks `/etc/modules.conf` and is the preferred way for loading modules. For detailed information on this topic, please refer to the corresponding manual page.

- **`lsmod`**
  Shows you which modules are currently loaded and by how many other modules they are being used. Modules started by the kernel daemon are tagged by `autoclean`, which shows that these modules will be removed automatically when they reach their idle time limit.

`/etc/modules.conf`

Loading of modules is further influenced by `/etc/modules.conf`. See also manpage for **`depmod`** (**`man depmod`**).

In this file, the parameters for modules which access hardware directly can be entered as such modules may need system-specific options (e. g. CD-ROM-driver or network driver). The parameters entered here are, in principle, identical to those given to the boot prompt of the kernel e. g. for LILO (see Section 10.3.2 page 247), but in many cases the names which are used at the boot prompt are different (to compare these, see Section 10.3.4 page 259). If a module failed to load, try specifying the hardware in this file and use modprobe to load the module instead of insmod.

Kmod – The "Kernel Module Loader"

From version 2.2.x, the kernel module loader is the most elegant way to use modules and replaces the old kernel daemon (kerneld). This kernel feature allows the kernel to launch modprobe directly and ensures that the necessary modules are loaded as soon as required by the kernel.

To use the kernel module loader, you must set the corresponding variable in the kernel configuration '`Kernel module loader`' (`CONFIG_KMOD`)

The drivers needed to access the root file system should be compiled directly into the kernel. So you should not configure your SCSI driver or your file system (normally: **`ext2`**) as modules!

Caution

Because SuSE Linux however, now uses `initrd` (initial ramdisk) and integrates the SCSI driver, for example, using this method, you must make sure when compiling your own kernel that you adjust the variable **`INITRD_MODULES`** in the file `/etc/rc.config` (Section 13.6 page 319) and comment out the `initrd` line in `/etc/lilo.conf` (see Section 12.4.5 page 297). If you don't do this the kernel will hang when you boot.

Kernel modules come in handy for rarely used functionality, such as `parport` and printer support, drivers for floppy drives and file systems that are rarely used.

Kmod is not designed to automatically unload modules; the potential saving in memory is only marginal for the RAM capacity of computers today; see also

/usr/src/linux/Documentation/kmod.txt. For reasons of performance, it is better for server machines which have special tasks to perform and need only a few drivers to have a "monolithic" kernel.

9.3 Kernel Configuration

The configuration of the kernel that was set up during installation or during an update can be taken from the file /usr/src/linux/.config.

Configuring the kernel can be done in three different ways:

1. On the command line
2. In a menu in text mode
3. In a menu in the X Window System

Here is a short overview of these three methods.

Configuring on the Command Line

To configure the kernel, just change to /usr/src/linux and enter:

```
earth:/usr/src/linux # make config
```

You are asked to choose the options that you want supported by the kernel you are about to build. There are two or three possible answers here: y, n or m. 'm' means that this device is not compiled directly into the kernel but as a module instead. Any driver that is needed to boot the system should be integrated into the kernel, and not be loaded as a module. If you press any other key, you get a short help text about the current option.

Configuring in Text Mode

A much more convenient way of configuring the kernel can be achieved by typing:

```
earth:/usr/src/linux # make menuconfig
```

With make menuconfig, you can review your changes, go through the questions in your own preferred order and, in the event of a mistake, you do not have to go through all the questions again.

Configuring in the X Window System

If you have installed and configured the X Window System (package xf86) and Tcl/Tk (package tcl and package tk), you can use:

```
earth:/usr/src/linux # make xconfig
```

as an alternative. You will be presented with a GUI (Graphical User Interface) which makes kernel configuration very user-friendly. You should have started the X Window System as 'root' or you will have to take additional steps into consideration (e. g., taking over the display from another user).

9.4 Settings in the Kernel Configuration

All the individual configuration possibilities of the kernel cannot be covered here in detail. Please make use of the numerous help texts available on kernel configuration. The latest kernel documentation is always in `/usr/src/linux/Documentation`.

9.5 Compiling the Kernel

Tip

You may remove the comment in the main `Makefile` (app. line 92). That's the line containing: `export INSTALL_PATH=/boot`. This lets you install your own kernel to `/boot`.

We recommend that you compile a "bzImage". As a rule this avoids the problem of the kernel getting *too large*, as can easily happen if you select too many features and create a "zImage" (messages such as `"kernel too big"` or `"System is too big"`) are then typical.

After adapting the kernel configuration to your needs, start compilation by entering:

```
    earth:/usr/src/linux # make dep
    earth:/usr/src/linux # make clean
    earth:/usr/src/linux # make bzImage
```

These three commands can be entered on one line as well. They are started one after the other. This might be useful if you want to compile a kernel overnight, for example. Just enter:

```
earth:/usr/src/linux # make dep clean bzImage
```

Depending on your system, it now takes just a few minutes (AMD Athlon / Intel Pentium III) up to several hours for a 386 with 8 MB.

While compiling, you can still work on one of the other consoles.

After a successful compilation, you will find the compressed kernel in `/usr/src/linux/arch/i386/boot` . The kernel-image – the file which contains the kernel – is called `bzImage`. If you cannot find this file, in all probability an error has occurred during the kernel compilation. This can easily be overlooked due to the large amount of screen output. If an error has taken place, this can be determined by setting off the kernel compilation again, with

```
earth:/usr/src/linux # make bzImage
```

and watching out for corresponding error messages. But don't worry: errors do not occur very often when the kernel is being compiled!

If you are using the Bash shell you may enter:

```
earth:/usr/src/linux # make bzImage 2>&1 | tee kernel.out
```

for writing the compilation output to `kernel.out`. Using the Tcsh this command is invoked as:

```
earth:/usr/src/linux # make bzImage |& tee kernel.out
```

If you have configured parts of your kernel to be loaded as modules, you need to launch the compilation of modules after you have compiled the kernel. This may be achieved by typing:

```
earth:/usr/src/linux # make modules
```

9.6 Installing the Kernel

After having compiled a kernel, you have to make sure that it is available for use from now on.

If you use LILO, then this must also be reinstalled. In the simplest case, copy the the new kernel to `/boot/vmlinuz` (see Section 9.5 on the preceding page) and then run LILO to protect yourself from unpleasant surprises, however, it is recommended that you initially retain the old kernel (`/boot/vmlinuz.old`), so that you can still boot it if the new kernel does not function as expected:

```
  earth:/usr/src/linux # cp /boot/vmlinuz /boot/vmlinuz.old
  earth:/usr/src/linux # cp arch/i386/boot/bzImage /boot/vmlinuz
  earth:/usr/src/linux # lilo
```

The Makefile target **`make bzlilo`** performs these three steps, by the way, in one go.

Now the compiled modules still need to be installed; by entering

```
  earth:/usr/src/linux # make modules_install
```

you can copy these to the correct target directories in `/lib/modules/<Version>`. In doing so, the old modules will be overwritten with the same kernel version; you can, however, reinstall the original modules together with the kernel from the CDs.

Tip

You should make sure that modules whose functionality may now have been compiled directly into the kernel are removed from `/lib/modules/<version>`. Otherwise unexpected effects could occur! This is one reason why the inexperienced user is *strongly* advised not to compile the kernel himself.

In addition, enter a label **`Linux.old`** as boot image in your `/etc/lilo.conf` (also see Section 4.4.1 page 99)and rename the old kernel `/boot/vmlinuz.old`. Thus you can make sure that you can still boot using the previous kernel if it does not end up working with the new one. This option is described in detail in Chapter 4 page 93.

When you have adapted `/etc/lilo.conf` to your needs, you can enter:

```
earth:/usr/src/linux # lilo
```

If you boot Linux via DOS using `linux.bat` (`loadlin`), you have to copy the kernel to `/dosc/loadlin/bzimage` (or to the directory where you have installed `loadlin`) for it to become active at the next boot.

If you start Linux via the Windows NT bootmanager you have to copy the LILO-bootsector to the Windows NT start partition again (see Section 4.7.2 page 109 for details).

Furthermore, the file /System.map contains kernel symbols which are needed by the modules to launch kernel functions correctly. This file depends on the current kernel. Therefore, you should copy /usr/src/linux/System.map to the directory (**/boot**). If you create your kernel using **make bzlilo**, this is done for you automatically.

If you get an error message like "System.map does not match current kernel", then probably System.map has not been copied.

9.7 Creating a Boot Disk

If you want to create a boot disk with the new kernel, you can use the following command:

```
earth:/usr/src/linux # make bzdisk
```

9.8 Cleaning Your Hard Disk After Compilation

If you are low on hard disk space, you can regain some of the hard disk space used during compilation, with:

```
earth: # cd /usr/src/linux
earth:/usr/src/linux # make clean
```

If there is plenty of space and you plan to compile a new kernel from the same sources, you might skip the last step. A new compilation will then be faster, as only those parts of the system which must change based on your choices are recompiled.

10 Kernel Parameters

10.1 Drivers in the Kernel

There is a wide variety of PC hardware components. In order to be able to use this hardware properly, you need a “driver” with which the operating system (in Linux, the “kernel”), can access this hardware. In general there are two ways of integrating drivers into your system:

1. The drivers can be compiled directly into the kernel. Such a kernel (“in one piece”) is referred to as a *monolithic* kernel. Some drivers are only available in this form, justifying the need for monolithic kernels.

2. Drivers can be loaded on demand into the kernel, which is then referred to as a *modularized* kernel. This has the advantage that only those drivers are loaded which really are needed, and the kernel thus contains no unnecessary ballast. Our SuSE boot disk kernel makes use of modules to support most hardware configurations.

Regardless of whether the drivers are compiled directly into the kernel or are loaded as modules, it is still possible that a hardware component may not be recognized by the kernel. If this is the case, you have the option of specifying component attributes more exactly.

For monolithic kernels these parameters must be given at the boot prompt or by means of a boot loader.[1] Modular drivers get their parameters via insmod or modprobe, which load the module itself simultaneously.

You always have to give the parameters, when you boot your computer, as the kernel does not learn. Later, after the installation, kernel parameters can be set in the files `/etc/lilo.conf` or `/etc/modules.conf`, so that LILO or modprobe can interpret them automatically.

Unfortunately, the format for parameters that are compiled directly into the kernel is different from that for parameters that are loaded on demand. For this reason they are divided into two different groups in the following pages. A few modules now have unified parameters (e. g., CD-ROM drives), so that the same parameters can be given whether you are using modules or the boot prompt.

[1] For this reason the parameters are also called LILO parameters, named after the time-honoured loader for X86 architectures

10.2 Some Tips

Before we present the list of parameters, here are some hints on recognizing hardware, adding parameters and booting with the SuSE boot disk:

- Most drivers can do an *autoprobing*, e. g., test different standard addresses where this hardware can usually be found. While autoprobing, a driver might initialize an address that does not belong to it. This can cause the machine to hang.
- Some modules can be loaded successfully, even if their hardware is not installed. This applies mainly to 3Com network card drivers. You should try autoprobing first. Unused drivers can be easily removed. Hardware that is not automatically recognized must then be started using the appropriate parameters.
- Finally, there are some hardware components that have more than one driver (e. g., NCR 53C810, Ultrastor). As far as we know, there is no significant difference between the two Ultrastor drivers. The BSD driver for NCR53C810 also supports other NCR53C8xx products (e. g., 53C875), whereas the old NCR driver is the only one that recognizes CD writers. Use the one that suits your needs.

10.3 The Parameters

10.3.1 Notations and Meanings

Below you will see an alphabetical list of kernel parameters and the corresponding devices. The following parameters always have the same meaning:

<addr>	hexadecimal port address, (e. g., 0x300)
<irq>	interrupt the device will use (e. g., 7)
<dma>	DMA channel of the device (e. g., 1)
<Start address>, <End address>	hexadecimal memory allocation for *shared memory*

Table 10.1: Frequently used variable names for kernel parameters

We are mainly concerned with the parameters that are required for a successful installation. But there are other parameters for special aims and purposes. Please keep in mind that upper case letters are significant.

A complete introduction to possible kernel parameters can be found after installation in the `BootPrompt-HOWTO`, in `/usr/share/doc/howto` as well as in the kernel sources in the file
`/usr/src/linux/Documentation/kernel-parameters.txt`.

10.3.2 Kernel Parameters at the Boot Prompt

The parameters listed in this section can only be entered to the kernel, for example, at the SYSLINUX (supplied bootdiskette), at the LILO-Prompt or via loadlin

If you plan to install one of these drivers as a module, please look at the parameters in Section 10.3.4 page 259.

> **Note**
>
> All parameters must be entered directly one after another, separated by commas. There should be no blanks (whitespace) between the parameters.

For example, to enter the parameters for the **`aha1542`** at the boot prompt you should enter (please keep in mind that you always have to precede the parameters with the boot configuration's name; on SuSE boot disks this is the configuration with the name `linux`!):

```
linux aha1542=0x300
```

General boot parameters

By means of certain parameters the general behavior of the Linux kernel can be controlled.

- *Reboot mode* (shutting down Linux)

  `reboot=<mode>`

  Here, you have the following values for <**`mode`**>:

Variable	Values / Meaning
`warm`	warm reboot (no memory check)
`cold`	cold reboot (with memory check)
`bios`	BIOS reboot
`hard`	CPU crash reboot (triple fault)

  Example: **`reboot=cold`**
  This boots the machine after shutdown as if the reset button had been pressed.

- *Protecting memory segments* (Reserve)

  `reserve=<start1>,<range1>,...,<startN>,<rangeN>`

  By means of this parameter, you can reserve memory space for hardware which does not like being autoprobed, and reacts with a system crash, for example.

  Example: A "sensitive" network card can be protected from being autoprobed and initialized by entering:

  `reserve=0x330,32 ether=5,0x330,eth0`

  In this example, the network card has a 32 bit data bus which starts at address `0x330` and uses interrupt `5`.

  For the full description of network card parameters, see Section 10.3.2 page 255.

- *Passing root partitions*

  ```
  root=<partition>
  ```

Variable	Values / Meaning
`<partition>`	e. g., `/dev/hda1`, `/dev/sdb5`

  Example: **`root=/dev/hda5`**
  This boots the kernel and tries to load the root partition from the first logical drive in the extended partition to the first (E)IDE hard disk.

- *Size of RAM* (RAM)

  ```
  mem=<size>
  ```

  You may enter the size of RAM in bytes, kilobytes or megabytes. Our example shows the different settings for 96 MB RAM.

  Example:

  ```
  mem=96M
  mem=98304k
  ```

  On some rare occasions it might happen that the mainboard cannot free the total amount of RAM. Please subtract the amount of memory the mainboard reserves (up to 512 K is considered normal). The exact amount can only be found by “trial and error”. Let’s assume that it is 512 KB for our example:

  ```
  mem=5ff8000
  ```

  When using a Pentium clone

  ```
  mem=nopentium
  ```

  this line might make it work.

- *Number of CPUs when using SMP*

  ```
  maxcpus=<number>
  ```

  Set the maximum number of CPUs to <number> when using an SMP kernel.

  Example:

  ```
  maxcpus=1
  ```

- *Serial console*

  ```
  console=<device>,<options>
  ```

  Set the device for the system console; this parameter can be used several times.

  Examples:

  ```
  console=ttyS1,9600
  console=tty0
  ```

  See `/usr/src/linux/Documentation/serial-console.txt` for more examples.

The SCSI controller and the SCSI subsystem

Most SCSI controllers can be configured via parameters.

- *SCSI Streamers*

  st=<buffer>,<threshold>[,<max>]

Variable	Values / Meaning
<**buffer**>	Buffer size (Amount of 1 KB block)
<**threshold**>	Write threshold (Amount of 1 KB blocks)
<**max**>	Maximum buffer size (optional) (e. g. 2)

  Example: **st=1000,2000**

- *Number of SCSI devices per ID*

  max_scsi_luns=<number>

Variable	Values / Meaning
<**number**>	`1..8`

  Example: If you want to use only the first LUN you need to set **max_scsi_luns=1**

  This is typically used for imperfect CD changers where the amount of usable CDs equals the parameter **max_scsi_luns=1**.

- *Adaptec AHA-1520 / 1522 / 1510 / 1515 / 1505 SCSI host adapter*

  aha152x=<addr>,<irq>,<id>[,<rec>[,<par>]]

Variable	Values / Meaning
<**id**> (SCSI ID of the host adapter)	`0,1`
<**rec**> (reconnect)	`0,1`
<**par**> (parity)	`0,1`

  This driver is used to run multiple low-cost SCSI controllers. All sound cards with a SCSI controller (except for Pro Audio Spectrum) can be accessed using the installed Adaptec chip.

  For all non-original 152x, the fourth parameter (**RECONNECT**) seems to be necessary. For most types, it has to be set to `‘0’`. Only the AHA2825 needs `‘1’`.

  Example: **aha152x=0x300,10,7**

- *Adaptec AHA-1540 / 1542 SCSI host adapter*

  aha1542=<addr>[,<buson>,<busoff>[,<DMA speed>]]

Variable	Values / Meaning
<**buson**>	`2..15`
<**busoff**>	`1..64`
<**DMA speed**>	`5,6,7,8,10`

  Example: **aha1542=0x300**

- *Adaptec AHA-274x / 284x / 294x Hostadapter*

  aic7xxx=<modifier>[,<modifier>[, ...]]

Variable	Values / Meaning

<**modifier**>	`extended`
	activates translation of hard disk geometry
	`no_reset`
	disables reset of the
	SCSI bus at the host adapter initialization
	`irq_trigger:<`**`x`**`>`
	Only for Eisa systems
	`0` for flange triggered,
	`1` for lever driven
	`verbose`
	receive more messages
	`reverse_scan`
	If multiple cards are treated in the wrong
	order by the BIOS
	`7895_irq_hack:<`**`x`**`>`
	`-1` only for Tyan II Motherboards
	`pci_parity:<`**`x`**`>`
	if `pci_parity` is not used at all
	parity is even
	`0` no parity check
	`1` parity odd
	`tag_info:„„„`
	Queue management for performance enhancement,
	for experts, see kernel sources

  Example: **aic7xxx=no_reset**,
  if the machine hangs while resetting the SCSI bus.

  parameters for aic7xxx based SCSI host adaptors are only necessary if they are faulty or malfunctioning.

  From BIOS version 1.3 onwards the AHA-2940 AU causes no trouble. Updates may be obtained from Adaptec support.

  The SCSI hostadapter Adaptec 2920 is driven by the Future Domain driver (see Section 10.3.2 on the facing page)

- *AdvanSys SCSI host adapter*

  advansys=<addr1>,<addr2>,...,<addrN>

  Example: **advansys=0x110,0x210**

  This example tells the kernel to search for the AdvanSys host adapter at the given addresses.

- *AM53/79C974 SCSI host adapter*

  AM53C974=<host-id>,<target-id>,<rate>,<offset>

Variable	Values / Meaning
<**host-id**>	the host adapter's SCSI ID (often 7)
<**target-id**>	the device's SCSI ID (often 0..7)
<**rate**>	3,5,10 MHz/s max. transfer rate
<**offset**>	transfer mode; 0 = asynchron

If the host adapter seems to "swallow" some packages, reduce the maximum transfer rate for this device (e. g., the first CD-ROM SCSI drive /dev/scd0 with ID 5) on the SCSI bus with:

Example: **AM53C974=7,5,3,0**

- *BusLogic SCSI host adapter*

  BusLogic=<**addr**>
  BusLogic=<**probing**>

  Example: **BusLogic=0x300**

Variable	Values / Meaning
<**addr**>	Address of the adapter, e. g. 0x300
<**probing**>	NoProbe No adaptor is probed NoProbeISA No ISA adapter is probed NoProbePCI No PCI adapter is probed NoSortPCI Order of multimaster adapter set by PCI BIOS MultiMasterFirst Multimaster before Flashpoint FlashPointFirst Flashpoint before Multimaster InhibitTargetInquiry For old devices that cause trouble with **scsi_luns** > 0 TraceProbe outputs additional messages at initialization of the adapter TraceHardwareReset outputs additional messages at the adaptor hardware reset TraceConfiguration outputs additional messages at the adaptor configuration TraceErrors outputs error messages of the attached devices Debug outputs all

  This host adapter understands even more parameters. These are for fine tuning the adaptor and are described in /usr/src/linux/drivers/scsi/README.BusLogic

- *Future Domain TMC-16x0 SCSI host adapter*

  fdomain=<**addr**>**,**<**irq**>**[,**<**id**>**]**

Variable	Values / Meaning
<**id**>	SCSI ID of the host adapter 0..7

This driver also controls the Adaptec 2920.

Example: **`fdomain=0x140,11,7`**

- *Future Domain TMC-885/950 host adapter*

  `tmc8xx=<addr>,<irq>`

  Example: **`tmc8xx=0xca000,5`**

- *NCR 5380 SCSI host adapter family*

  `ncr5380=<addr>,<irq>,<dma>`

  Example: **`ncr5380=0x340,10,3`**

- *NCR 53c400 SCSI host adapter family*

  `ncr53c400=<addr>,<irq>`

  Example: **`ncr53c400=0x350,5`**

- *NCR 53c406a SCSI host adapter family*

  `ncr53c406a=<addr>[,<irq>[,<fastpio>]]`

Variable	Values / Meaning
<**fastpio**>	0, if no fast PIO mode is required

  Example: **`ncr53c406a=0x330,10,0`**

- *Seagate ST01/02 SCSI host adapter*

  `st0x=<addr>,<irq>`

  Example: **`st0x=0xc8000,5`**

- *Trantor T128/128F/228 SCSI host adapter*

  `t128=<addr>,<irq>`

  Example: **`t128=0x340,10`**

(E)IDE controllers and ATAPI devices

Numerous parameters are available to configure the (E)IDE controllers and the devices connected to them.

- *ATAPI CD-ROM* on the (E)IDE controller

  ```
  hd<x>=cdrom
  hd<x>=serialize
  ```

Variable	Values / Meaning
<**x**>	a, b, c, d

where:

Variable	Values / Meaning
a	master on 1st IDE controller
b	slave on 1st IDE controller
c	master on 2nd IDE controller
d	slave on 2nd IDE controller

Example: an ATAPI CD-ROM as master on the secondary IDE controller is set by **`hdc=cdrom`**.

- *Hard Disk*

  `hd<x>=<cylinders>,<heads>,<sectors>[,<write>[,<irq>]]`

Variable	Values / Meaning
<**`x`**>	`a`, `b`, . . . , `h` 1 to 8. Hard Disk
<**`cylinders`**>	number of cylinders
<**`heads`**>	number of heads
<**`sectors`**>	number of sectors
<**`write`**>	cylinders after which write compensation is used
<**`irq`**>	interrupt

  If the BIOS is an old one, it is possible that the geometry of the hard drive is not recognized correctly. The correct parameters will then be passed on so that the kernel can still access the complete hard disk.

  Example: **`hdc=1050,32,64`**

  `hd<x>=<trouble>`

Variable	Values / Meaning
<**`x`**>	`a`, `b`, . . . , `h` 1st to 8th hard disk
<**`trouble`**>	`noprobe`, if testing an existing hard disk causes problems `none` ignore CMOS entry and don't test `nowerr` ignore WREE_STAT-bit `cdrom` wrongly recognized as hard disk or not recognized at all, or will not boot `autotune` the fastest PIO mode is used `slow` adds a long break after each access. This makes it really slow, but it sometimes helps if it is the only option left

  If a CD-ROM drive is not recognized reliably, specifying <**`cdrom`**> can register the device safely.

  Example: **`hdd=cdrom`**

- *EIDE controller chipsets*

Some EIDE controllers have faulty chipsets or cause problems if the secondary controller is used.

Many of these chipsets are now supported in the kernel; this support needs to be specially activated, however, using a kernel parameter.

The following chipsets can be configured:

`CMD 640`	This chipset is found on many motherboards. Since it contains many bugs, the kernel offers a special support which recognizes the chip and bypasses the problems. Moreover, in some cases, use of the secondary controller is only possible by using this special support. In PCI systems, this chip is automatically recognized. For VLB systems, the following parameter is needed: **`ide0=cmd640_vlb`**.
`RZ 1000`	This chip is used on many motherboards that use the Neptune chipset and it is buggy. If support for this chip is activated, the system works a little slower but reliably. An additional activation with a kernel parameter is not required.
`DTC-2278`	Only activating this driver via **`ide0=dtc2278`** makes it possible to use the secondary controller.
`Holtek HT6560B`	The following parameter is needed to activate the secondary controller: **`ide0=ht6560b`**.
`QDI QD6580`	If this driver is activated, it enables a higher speed: **`ide0=qd6580`**.
`UMC 8672`	For activating the secondary controller, the following parameter is needed: **`ide0=umc8672`**.
`ALI M1439/M1445`	For activating the secondary controller, the following parameter is needed: **`ide0=ali14xx`**.
`PROMISE DC4030`	For activating the secondary controller, the following parameter is needed: **`ide0=dc4030`**. CD-ROMs and tapes on the secondary controller are not supported yet.

Table 10.2: Special EIDE chipsets

If the chipset does not belong to the list of known faulty chipsets, and still is not recognized, the following parameters can be added instead:

```
ide<number>=<basis>[,<control>[,<irq>]]
```

Variable	Values / Meaning
<**number**>	adapter number, usually `0` or `1`, but also `3` or `4`
<**basis**>	base address of the adapter, usually `0x1f0`, `0x170`, `0x1e8` or `0x168`
<**control**>	control register of the adapters, usually `0x3f6`, `0x376`, `0x3ee` or `0x36e`
<**irq**>	interrupt of the adapters, usually `14`, `15`, `11` or `10`

If the chipset does not belong to the list of known faulty chipsets, but still causes problems, the following parameters can be added instead:

ide<**number**>**=**<**tune**>

Variable	Values / Meaning
<**number**>	adapter number, usually `0` or `1`, but also `3` or `4`
<**tune**>	`autotune` the highest possible PIO value is tried, not supported by all chipsets `noautotune` no improvement in speed `serialize` no time overlapping of operations with the next adapter

If the chipset does not belong to the list of known faulty chipsets, but the highest possible speed should still be achieved, the bus speed can be passed on; please refer to your motherboard manual.

idebus=<**speed**>

Other devices

- *Ethernet network cards*

  ether=<**irq**>**,**<**addr**>**[,**<**par1**>**[,**<**par2**>**...**<**par8**>**]],**<**Name**>

  The various parameters for <**par1**> to <**par8**> have different meanings for several drivers. Usually, only two parameters are needed, where the first is the start address and the second is the end address of *shared memory*. The first non-numerical argument is treated as the name.

<**irq**>	the interrupt used; `0` for autoprobing
<**addr**>	port address; `0` for autoprobing
<**start**>	start address for shared memory. Some drivers use the 4 lowest bits for the debug level. The Lance uses them for its DMA channel.
<**end**>	end address for shared memory. The 3COM 3c503 driver uses this parameter to distinguish between internal and external transceivers.

Table 10.3: continued overleaf...

	The Cabletron E21XX card uses the lowest four bits for selecting the media.
<**Name**>	The interface's name (normally **eth0**)

Table 10.3: Variable names for Ethernet network cards

This parameter lets the kernel recognize more than one network card, since only the first card is searched by default. This can easily be done with:

```
ether=0,0,eth1
```

By passing `0` for both IRQ and address, the driver is told to launch *autoprobing*, which means probing several addresses independently.

- *Floppy disk drives*

  floppy=<drive>,<type>,cmos

Variable	Values / Meaning
<**drive**>	0, 1, 2, 3
<**type**>	`0` - unknown or not recognized `1` - 5 1/4" DD, 360 KB `2` - 5 1/4" HD, 1.2 MB `3` - 3 1/2" DD, 720 KB `4` - 3 1/2" HD, 1.44 MB `5` - 3 1/2" ED, 2.88 MB `6` - 3 1/2" ED, 2.88 MB

floppy=<value>

For <**value**>, the following values may be set (Table 10.4 on the next page):

all_drives	more than two floppy disk drives
asus_pci	denies access to 3rd and 4th floppy disk drives
daring	use only with reliable controllers—enhances performance
0,daring	opposite of **daring**
<**addr**>**,two_fdc**	if the value <**addr**> is omitted while using a secondary floppy controller, a port address of `0x370` is set.
thinkpad	IBM Thinkpad machines
0,thinkpad	not a Thinkpad machine
omnibook	Omnibook computers
nodma	for Omnibook computers
dma	Standard

Table 10.4: continued overleaf...

nofifo	if a `"Bus master arbitration error"` occurs
fifo	Standard
`0xX`**,fifo_depth**	FIFO threshold standard `0xA`
unexpected_interrupts	show warnings if something unexpected happens.
no_unexpected_interrupts	and...
L40SX	this value is the opposite of **unexpected_interrupts**.

Table 10.4: Kernel parameter values for **floppy**

- *Logitech bus mouse*

  bmouse=<**irq**>

10.3.3 CD-ROM Drives on Proprietary Controllers

- *Aztech CDA268-01 CD-ROM*

  aztcd=<**addr**>**[,0x79]**

  The value `0x79` has only to be set when the firmware version is unknown.

  Example: **aztcd=0x320**

- *Goldstar R420 CD-ROM drive*

  gscd=<**addr**>

- *Mitsumi CD-ROM drive*

  mcd=<**addr**>**,**<**irq**>**[,**<**wait**>**]**

Variable	Values / Meaning
<**wait**>	value for waiting time while starting

  You can vary the parameter <**wait**> between `0` and `10` if the CD-ROM drive does not react fast enough on system requests (`"timeout"`) and, therefore, does not find the root image while installing.

  Example: **mcd=0x300,10,5**

- *Mitsumi CD-ROM drive (multisession)*

  mcdx=<**addr**>**,**<**irq**>

  Example: **mcd=0x300,10**

- *Mozart interface*

  isp16=<**addr**>**,**<**irq**>**,**<**dma**>**,**<**type**>

Variable	Values / Meaning
<**type**>	Sanyo, Panasonic, Sony, Mitsumi

This driver is responsible for CD-ROM drives connected to the types ISP16, MAD16 or Mozart. The value of <**type**> follows the interface plug where the CD-ROM is connected to the sound card.

Example: **`isp16=0x340,10,3,Sony`**

- *Optics Storage 8000 AT CD-ROM drive*

  `optcd=<addr>`

  Example: **`optcd=0x340`**

- *Philips CM206 CD-ROM drives*

  `cm206=<addr>,<irq>`

  Example: **`cm206=0x340,10`**

- *Pro Audio Spectrum 16 - SCSI host adapter*

  `pas16=<addr>,<irq>`

  On the Pro Audio Spectrum 16 sound card there is a SCSI-host adapter, the settings of which can be passed to the kernel with this parameter.

  Example: **`pas16=0x340,10`**

  If the card is not recognized, it can be operated without an interrupt. In this case the interrupt is set to `255`.

  Example: **`pas16=0x340,255`**

- *Sanyo CD-ROM Drive*

  `sjcd=<addr>`

  Example: **`sjcd=0x340`**

- *Sony CDU 31/33 A*

  `cdu31a=<addr>,<irq>`

  Autoprobing of this driver has been removed from the kernel. Thus, you have to set any parameters explicitly.

  Example: **`cdu31a=0x340,5`**

  If there is no interrupt assigned to the drive, and if you can only access it via polling mode, you have to enter `0` for the IRQ.

  Example: **`cdu31a=0x340,0`**

  As an example, if this drive is connected to a Pro Audio Spectrum card, the parameter could be as follows:

  Example: **`cdu31a=0x1f88,0,PAS`**

- *Sony CDU 535*

  sonycd535=<addr>,<irq>

  Example: **sonycd535=0x340,10**

- *SoundBlaster Pro 16 MultiCD*

  sbpcd=<addr>,<type>

Variable	Values / Meaning
<**type**>	`LaserMate`, `SPEA`, `SoundBlaster`

  Example: **sbpcd=0x340,10**

The parallel port

- *Parallelport*

  parport=<addr0>,[<irq0> [parport=<addr1>,[<irq1> [parport=<addr2>,[<irq2>]]]]]

Variable	Values / Meaning
<**addrX**> <**irqX**>	address interrupt

  Example: **parport=0x3bc parport=0x378,7 parport=0x278,auto**

- *Printer on the parallel port*

  lp=<parport0> [lp=<parport1> [lp=<parport2>]]

Variable	Values / Meaning
<**parportX**>	parallel port

  Example: **lp=parport0 lp=parport2**

10.3.4 `modprobe` Parameters

This section describes those parameters which can be loaded as modules. If you encounter difficulties while loading a driver (although you have entered the parameters), or if there is no section describing the parameter, then you may have to integrate this driver into a monolithic kernel.

Some drivers cannot be built as modules yet and some recognize your hardware properly only if they are compiled into the kernel. Despite this, we recommend you try the “module variant” first.

If a driver is loaded as a module, each and every variable used can be overwritten on the command line. There is, e. g., the variable **io** in the NE2000 driver which specifies the used I/O range. For this, the correct command for loading this module is (see Section 9.2 page 238):

```
earth:/ # insmod ne io=0x300 irq=10
```

or better with modprobe:

```
earth:/ # modprobe ne io=0x300 irq=10
```

Be aware that there should be no spaces before or after the “equal”. Moreover, hexadecimal values have to be set in the given form (with a leading `'0x'`).

> **Note**
>
> If you want to enter more than one parameter they must be separated by spaces. This is the main difference to entering parameters at the LILO prompt, where no blanks should be used within the parameters for one driver.

The parameters you enter here can also be integrated into `/etc/modules.conf`. Here many parameters can be assigned to a particular module. This is done one line per module. The line should resemble the following:

```
options <module name> <parm1>=<value1> ...
```

where:

Variable	Values / Meaning
<**module name**>	name of the module without the extension `.o`
<**parm1**>	parameter #1
<**value1**>	value that is assigned to parameter #1

An entry for the NE2000 card might look like:

```
options ne io=0x300 irq=10
```

SCSI controllers and the SCSI subsystem

- *Adaptec AHA-1520 / 1522 / 1510 / 1515 / 1505 SCSI-Hostadapter*

  Name of module: `aha152x.o`

Variable	Values / Meaning
<**io**>	<**addr**>
<**irq**>	<**irq**>
<**id**>	SCSI-ID of the host adapter; by default, `7`
<**rec**>	reconnect; `0`, `1`
<**par**>	parity; `0`, `1`
<**sync**>	synchronous mode of operation; `0`, `1`
<**delay**>	bus delay; default 100
<**translat**>	"translation of the hard disk geometry; `0`, `1`

As at the LILO prompt:
Example: modprobe aha152x aha152x=0x340,10,7,1,1,0,0,0

For a second host adapter:
Example: **`modprobe aha152x aha152x1=0x140,12,7,1,1,0,0,0`**

- *Future Domain TMC-16x0–host adapter*

  Name of module: `fdomain.o`

  This driver also serves the SCSI-host adapter Adaptec 2920, amongst others.

  `fdomain=<addr>,<irq>[,<id>]`

Variable	Values / Meaning
<**addr**>	<**addr**>
<**irq**>	<**irq**>
<**id**>	<**id**>

As at the boot prompt:
Example: modprobe fdomain fdomain=0x140,11,7

- *NCR 5380 bzw. NCR 53C400 SCSI-host adapter family*

  Name of module: `g_NCR5380.o`

Parameter	Value
ncr_addr	<**addr**>
ncr_irq	<**irq**>; switch off with 255
ncr_dma	<**dma**>
ncr_5380	1 for an NCR5380 adapter
ncr_53c400	1 for an NCR53C400 adapter

For an NCR5380 adapter:
Example: modprobe g_NCR5380 ncr_irq=5 ncr_addr=0x350
`ncr_5380=1`

For an NCR53C400 adapter with interrupts turned off:
Example: modprobe g_NCR5380 ncr_irq=255 ncr_addr=0xc8000
`ncr_53c400=1`

This driver serves the widely-used Trantor T130B SCSI host adapter

Network cards

- *3Com 3c501 / 3c503 / 3c505 / 3c507 network cards*

  Name of module: `3c501.o`, `3c503.o`, `3c505.o`, `3c507.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: modprobe 3c509 io=0x300 irq=10

- *3Com 3c509 / 3c579 network cards*

  Name of module: `3c509.o`

Parameter	Value
`irq`	<**`irq`**>
`xcvr`	`0`: internal; `1`: external

  Example: **`modprobe 3c509 irq=10 xcvr=0`**

- *3Com 3c515 network card*

  Name of module: `3c515.o`

  Example: **`modprobe 3c515`**

- *3Com 3c59x / 3c90x network cards ("Vortex"/"Boomerang")*

  Name of module: `3c59x.o`

  For Compaq computers(PCI), you can try the following parameters:

Parameter	Value
`compaq_ioaddr`	<**`addr`**>
`compaq_irq`	<**`irq`**>
`compaq_prod_id`	<**`id`**>

  Example: **`modprobe 3c59x compaq_irq=10`**

- *Allied Telesis AT1700 network card*

  Name of module: `at1700.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>

  Example: **`modprobe at1700 io=0x300 irq=10`**

- *Cabletron E21xx network card*

  Name of module: `e2100.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>
`mem`	<**`addr`**>
`xcvr`	`0`: internal; `1`: external

  Example: **`modprobe e2100 io=0x300 irq=10 mem=0xd000 xcvr=0`**

- *Digital DE425 / 434 / 435 / 450 / 500 network cards*

  Name of module: `de4x5.o`

  `io=0x`<**`bus`**><**`device ID`**>

Parameter	Value
`bus`	number of the PCI bus, usually `0`
`device ID`	number des PCI device

  With newer PCI BIOSes, these files are displayed when the machine boots, and in Linux they can be seen with the command

  ```
  earth: # cat /proc/pci
  ```

  Example: **`modprobe de4x5 io=0x007`**

- *Digital DEPCA / DE10x / DE20(012) / DE42, EtherWORKS network cards*

  Name of module: `depca.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>
`mem`	<**`mem`**>
`adapter_name`	<**`name`**> e.g. `DEPCA`, `de100`, `de101`, `de200`, `de201`, `de202`, `de210`, `de422`

  Example: **`modprobe depca io=0x300 irq=10`**

- *EtherWORKS 3 (DE203, DE204, DE205) network card*

  Name of module: `ewrk3.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>

  Example: **`modprobe ewrk3 io=0x300 irq=10`**

- *Intel EtherExpress 16 network card*

  Name of module: `eexpress.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>

  Example: **`modprobe eexpress io=0x300 irq=10`**

- *Intel EtherExpressPro network card*

  Name of module: `eepro.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**addr**>

  Example: **`modprobe eepro io=0x300 irq=10 mem=0xd000`**

- *Intel EtherExpressPro 100 network card*

  Name of module: `eepro100.o`

  The Intel EtherExpressPro contains the chips i82557/i82558.

Parameter	Value
options	<**duplex operation**> <**bus-transceiver**> `16` full-duplex `32` 100 MBit operation only `64` 10 MBit operation only

  Example: **`modprobe eepro100 options=48`**

  This sets *full-duplex* and *100-MBit operation* simultaneously ($48 = 32 + 16$).

- *Fujitsu FMV-181/182/183/184 network cards*

  Name of module: `fmv18x.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **`modprobe fmv18x io=0x300 irq=10`**

- *HP PCLAN+ (27247B and 27252A) network card*

  Name of module: `hp-plus.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **`modprobe hp-plus io=0x300 irq=10`**

- *HP PCLAN (27245 / 27xxx)*

  Name of module: `hp.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe hp io=0x300 irq=10`**

- *HP 10/100 VG-AnyLAN (ISA, EISA, PCI) network cards*

  Name of module: `hp100.o`

Parameter	Value
hp100_port	<**addr**>

Example: **`modprobe hp100 hp100_port=0x300`**

- *ICL EtherTeam 16i / 32 network cards*

  Name of module: `eth16i.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe eth16i io=0x300 irq=10`**

- *Novell NE2000 / NE1000 network cards*

  Name of module: `ne.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
bad	`bad`, only if the card is not recognized

Example: **`modprobe ne io=0x300 irq=10`**

- *NI6510 (AM7990 “lance” Chip) network card*

  Name of module: `ni65.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>

Example: **`modprobe ni65 io=0x300 irq=10`**

- *SMC Ultra network card*

  Name of module: `smc-ultra.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **`modprobe smc-ultra io=0x300 irq=10`**

- *SMC 9194 network card*

  Name of module: `smc9194.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
if_port	<**medium**>

Variable	Values / Meaning
<**medium**>	0 auto 1 TP 2 AUI, 10base2

  Example: **`modprobe smc9194 io=0x300 irq=10 if_port=2`**

- *Western Digital WD80x3 network card*

  Name of module: `wd.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**mem**>
mem_end	<**mem_end**>

  Example: **`modprobe wd io=0x300 irq=10`**

- *IBM Tropic chipset Token Ring network card*

  Name of module: `ibmtr.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem_start	<**mem_start**>

  Example: **`modprobe ibmtr io=0x300`**

- *D-Link DE620 Pocket-Adapter network card*

  Name of module: `de620.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
bnc	`1` if a BNC input/output
utp	`1` if a UTP input/output
clone	`1` if an equivalent device

  Example: **`modprobe de620 io=0x300 irq=10 bnc=1 utp=0`**

Proprietary CD-ROM drives

The following parameters concern CD-ROM drives attached to special controllers. Whoever possesses one of these “oldies” will certainly know about it…

- *Aztech CDA268-01 CD-ROM drive*

  Name of module: `aztcd.o`

Parameter	Value
aztcd	<**addr**>

  Example: **`modprobe aztcd aztcd=0x300`**

- *Goldstar R420-CD-ROM drive*

  Name of module: `gscd.o`

Parameter	Value
gscd	<**addr**>

  Example: **`modprobe gscd gscd=0x300`**

- *Mitsumi CD-ROM drive*

  Name of module: `mcd.o`

  mcd=<addr>,<irq>[,<wait>]

Variable	Values / Meaning
<**wait**>	Delay at startup time

  You can vary the <**wait**> from `0` to `10` if the CD-ROM drive does not react fast enough (`"timeout"`), thus probably not finding the root image during installation.

  Example: **`modprobe mcd mcd=0x300,10,5`**

- *Mitsumi CD-ROM drive (Multisession)*

  Name of module: `mcdx.o`

  mcdx=<addr>,<irq>[,<addr>,<irq>]

  For a number of drivers the settings <addr>, <irq> can be repeated up to 4 times.

- *Mozart sound card with port for CD-ROM drives*

  Name of module: `isp16.o`

  isp16_cdrom_base=<addr>
  isp16_cdrom_irq=<irq> isp16_cdrom_dma=<dma>
  isp16_cdrom_type=<typ>

Variable	Values / Meaning
<typ>	Sanyo, Panasonic, Sony, Mitsumi

  This driver is not a real CDROM driver; it is only responsible for the interface configuration of the CD-ROM drive when connected to a ISP16, MAD16 or Mozart sound card. After this driver is loaded only the interface is loaded accordingly; the corresponding CDROM driver needs to be loaded as well. The value for the variable <**type**> is written on the plug which connects the CD-ROM cable to the sound card.

  Example: **modprobe isp16 isp16_cdrom_base=0x300 isp16_cdrom_irq=10 isp16_cdrom_dma=1 isp16_cdrom_type=sony**

- *Optics Storage 8000 AT CD-ROM drive*

  Name of module: `optcd.o`

Parameter	Value
optcd	<**addr**>

  Example: **modprobe optcd optcd=0x300**

- *Philips CM206 CD-ROM drive*

  Name of module: `cm206.o`

  cm206=<addr>,<irq>

  Example: **modprobe cm206 cm206=0x300,10**

- *Sanyo CD-ROM drive*

  Name of module: `sjcd.o`

Parameter	Value
sjcd	<**addr**>

  Example: **modprobe sjcd sjcd=0x300**

- *Sony CDU 31/33 A*

  Name of module: `cdu31a.o`

Parameter	Value
cdu31a_port	<**addr**>
cdu31a_irq	<**irq**>

Example: **`modprobe cdu31a cdu31a_port=0x300 cdu31a_irq=10`**

- *Sony CDU 535*

  Name of module: `sonycd535.o`

Parameter	Value
sonycd535	<**addr**>

Example: **`modprobe sonycd535 sonycd535=0x300`**

- *Soundblaster Pro 16 MultiCD*

  Name of module: `sbpcd.o`

  sbpcd=<**addr**>**,**<**typ**>

  whereby <**type**> can have the following values:

Variable	Values / Meaning
0	`LaserMate`
1	`SoundBlaster`
2	`SoundScape`
3	`Teac16bit`

Example: **`modprobe sbpcd sbpcd=0x300,0`**

Sound cards and sound chipsets

Since the 2.2.xx kernel sound support has been resolved in modular form and can be controlled by numerous parameters.

- *AD1816 Chip*

  Name of module: `ad1816.o`

  The following cards, amongst others, are supported: Terratec Base 1, Terratec Base 64, HP Kayak, Acer FX-3D, SY-1816, Highscreen Sound-Boostar 32 Wave 3D

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
ad1816_clockfreq	<**clockfreq**>

Example: **`modprobe ad1816 io=0x530 irq=5 dma=1 dma2=3 ad1816_clockfreq=33000`**

- *AD1848/CS4248 Chip (MSS)*

  Name of module: `ad1848.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>
`dma`	<**`dma`**>
`dma2`	<**`dma2`**>
`type`	<**`cardtype`**>
`deskpro_xl`	<**`magic`**>

- *Generic OPLx driver*

  Name of module: `adlib_card.o`

Parameter	Value
`io`	<**`addr`**>

Example: **`modprobe adlib_card io=0x330`**

- *Crystal 423x chipsets*

  Name of module: `cs4232.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>
`dma`	<**`dma`**>
`dma2`	<**`dma2`**>

Example: **`modprobe cs4232 io=0x530 irq=5 dma=1 dma2=3`**

- *Ensoniq 1370 chipset*

  Name of module: `es1370.o`

  see also PCI64/128.

Parameter	Value
`joystick`	`1`: activate joystick
`lineout`	`1`: line-in in line-out switch over
`micz`	microphone impedancy

Example: **`modprobe es1370 joystick=1 lineout=1`**

- *Creative Ensoniq 1371 chipset*

  Name of module: `es1371.o`

  Refer also to the PCI64/128.

Parameter	Value
`joystick`	<**`addr`**>

  Valid values for <**`addr`**> are `0x200`, `0x208`, `0x210` and `0x218`.

  Example: **`modprobe es1371 joystick=0x200`**

- *Gravis Ultrasound*

  Name of module: `gus.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>
`dma`	<**`dma`**>
`dma16`	<**`dma16`**>
`type`	
`gus16`	
`no_wave_dma`	
`db16`	

- *MAD16*

  Name of module: `mad16.o`

  Supports OPTi 82C928, OAK OTI-601D, OPTi 82C929, OPTi 82C930 and OPTi 82C924.

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>
`dma`	<**`dma`**>
`dma16`	<**`dma2`**>

  Example: **`modprobe mad16 io=0x530 irq=7 dma=0 dma16=1`**

- *Turtle Beach Maui and Tropez*

  Name of module: `maui.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>

  Example: **`modprobe maui io=0x530 irq=5`**

- *MPU401*

  Name of module: `mpu401.o`

Parameter	Value
io	<**addr**>

  Example: **`modprobe mpu401 io=0x330`**

- *Turtle Beach MultiSound*

  Name of module: `msnd.o`

  Example: **`modprobe msnd`**

- *Turtle Beach Classic/Monterey/Tahiti*

  Name of module: `msnd_classic.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	
write_ndelay	
major	
fifosize	
calibrate_signal	

  Example: **`modprobe io=0x290 irq=7 mem=0xd0000`**

- *Turtle Beach Pinnacle/Fiji*

  Name of module: `msnd_pinnacle.o`

  As well as the parameters of the Turtle Beach Classic/Monterey/Tahiti, this card also understands:

Parameter	Value
digital	
cfg	
reset	
mpu_io	
mpu_irq	
ide_io0	
ide_io1	
ide_irq	
joystick_io	

  Example: **`modprobe msnd_pinnacle cfg=0x250 io=0x290 irq=5 mem=0xd0000`**

- *OPL3*

  Name of module: `opl3.o`

Parameter	Value
io	<**addr**>

Example: **`modprobe io=0x388`**

- *OPL3-SA1*

  Name of module: `opl3sa.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
mpu_io	<**addr**>
mpu_irq	<**irq**>

Example:

```
modprobe opl3sa io=0x530 irq=11 dma=0 dma2=1
mpu_io=0x330 mpu_irq=5
```

- *YMF711, YMF715, YMF719, OPL3-SA2, OPL3-SA3, OPL3-SAx*

  Name of module: `opl3sa2.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
mss_io	<**addr**>
mpu_io	<**addr**>

Example:

```
modprobe opl3sa2 io=0x370 irq=7 dma=0 dma2=3
mss_io=0x530 mpu_io=0x330
```

- *Pro Audio Spectrum*

  Name of module: `pas2.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma16	<**dma16**>
sb_io	<**addr**>
sb_irq	<**irq**>
sb_dma	<**dma**>
sb_dma16	<**dma16**>
joystick	
symphony	
broken_bus_clock	

- *Personal Sound System (ECHO ESC614)*

  Name of module: `pss.o`

Parameter	Value
pss_io	<**addr**>
mss_io	<**addr**>
mss_irq	<**irq**>
mss_dma	<**dma**>
mpu_io	<**addr**>
mpu_irq	<**irq**>
pss_mixer	activate `1` or `0`

- *Sound Blaster and Clones*

  Name of module: `sb.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma16	<**dma16**>
mpu_io	<**addr**>
mad16	activate `1`
trix	activate1
pas2	activate1
sm_games	activate1
acer	activate1, for Acer Notebooks
mwave_bug	activate1

Example: **`modprobe sb io=0x220 irq=5 dma=1 dma16=5 mpu_io=0x330`**

- *Aztech Sound Galaxy*

Name of module: `sgalaxy.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
sgbase	

- *S3 Sonic Vibes*

  Name of module: `sonicvibes.o`

  No parameters.

- *Ensoniq SoundScape*

  Name of module: `sscape.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
mss	
mpu_io	<**addr**>
mpu_irq	<**irq**>
spea	1

- *MediaTrix AudioTrix Pro*

  Name of module: `trix.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
sb_io	<**addr**>
sb_irq	<**irq**>
sb_dma	<**dma**>
mpu_io	<**addr**>
mpu_irq	<**irq**>

- *UART401*

  Name of module: `uart401.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe io=0x330 irq=9`**

- *UART6850*

  Name of module: `uart6850.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

- *Sound Blaster DSP chipsets*

  Name of module: `v_midi.o`

  No parameters.

- *Turtle Beach Maui, Tropez, Tropez Plus*

  Name of module: `wavefront.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **`modprobe wavefront io=0x200 irq=9`**

The parallel port

The operation of the parallel port is somewhat complex. After configuration, a complete subsystem is available; see Section 6.4.3 page 179 pp .

- *Parallel port*

  Name of module: `parport.o`

  Example: **`modprobe parport`**

- *Parallel port – architecture-specific*

  Name of module: `parport_pc.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  In order to set up 3 Ports in a PC style, for example, whereby one has the address `0x3bc` without an IRQ, one has the address `0x378` with IRQ 7 and one has the address `0x278` with automatically detected IRQ, you should enter (all in one line!):

  Example:
  ```
  modprobe parport_pc io=0x3bc,0x378,0x278
  irq=none,7,auto
  ```

- *Printer driver*

  Name of module: `lp.o`

Parameter	Value
`parport`	<**`port`**>

  Example: **`modprobe lp parport=0,2`**

- *Parallel port IDE devices*

  Name of module: `paride.o`

  If you want to use chains of Paride devices on a *single* parallel port, then you must first load `parport.o`!

  Example: **`modprobe paride`**

  The parallel port should – if possible – be operated in "EPP mode"; please set this mode in the BIOS of your computer.

- *Parallel port IDE low-level protocol driver*

  Name of module: `\variable{xxxx}.o`

protocol	device	
`aten`	ATEN EH-100	(HK)
`bpck`	Microsolutions backpack	(US)
`comm`	DataStor (old-type) "commuter" adapter	(TW)
`dstr`	DataStor EP-2000	(TW)
`epat`	Shuttle EPAT	(UK)
`epia`	Shuttle EPIA	(UK)
`fit2`	FIT TD-2000	(US)
`fit3`	FIT TD-3000	(US)
`friq`	Freecom IQ cable	(DE)
`frpw`	Freecom Power	(DE)
`kbic`	KingByte KBIC-951A and KBIC-971A	(TW)
`ktti`	KT Technology PHd adapter	(SG)
`on20`	OnSpec 90c20	(US)
`on26`	OnSpec 90c26	(US)

  Example: **`modprobe epat`**

- *Parallel port IDE hard disk*

  Name of module: `pd.o`

Parameter	Value
`verbose`	<**`value`**>

  First load `parport` and the low-level driver

  Example: **`modprobe pd verbose=1`**

- *Parallel port ATAPI CD-ROM*
  Name of module: `pcd.o`
  First load `parport` and the low-level driver
  Example: **`modprobe pcd`**

- *Parallel port ATAPI floppy disk drive*
  Name of module: `pf.o`
  First load `parport` and the low-level driver
  Example: **`modprobe pf`**

- *Parallel port ATAPI streamer drive*
  Name of module: `pt.o`
  First load `parport` and the low-level driver
  Example: **`modprobe pt`**

- *Parallel port generic ATAPI device*
  Name of module: `pg.o`
  First load `parport` and the low-level driver
  Example: **`modprobe pg`**

11 Updating the System and Package Management

11.1 Updating SuSE Linux

SuSE Linux offers you the option of updating an existing system without having to reinstall everything. It is important to distinguish between updating one or a few packages and updating the entire system to the latest distribution.

It is quite normal for software to "grow". Thus we recommend that you have a look at how full the partitions are, with **`df`**, *before* updating! If you think there might be too little space available, you should consider making a backup and repartitioning. There is no rule of thumb as to how much space you need in particular cases.

Note

It is helpful to read the file `README` or in DOS/Windows `README.DOS` on the CD; here additional changes are noted which may have been made *after* the manuals have been printed.

11.1.1 Preparations

Before you begin an update, make sure you have saved the old configuration files to a separate medium such as a streamer, removable drive, floppies or ZIP drives, just in case things go wrong. Normally these are the files in `/etc` and `/var/lib` (e. g. XDM). In addition, no harm would be done if you wrote the current user data in `/home` (the `HOME` directories) to a backup medium. A data backup should be carried out as System administrator `'root'`; only `'root'` has the required permissions to read all local files.

Before you start the update process, remember to write down the device name of your root partition. In this case, `/dev/sda3` would be your root partition. You can see this with the command:

```
earth: # df /

Filesystem  1024-blocks  Used Available Capacity Mounted on
/dev/sda3      4167999 3253471   698856     82%   /
/dev/sda1         7496    1311     5785     18%   /boot
```

Output 11.1.1: file system overview with **`df`**

This shows that /dev/sda3 is mounted in the file system in /.

> Tip
>
> Before updating *PostgreSQL* (package postgres) we recommend that you make a dump of the databases (see manpage for **pg_dump** (**man pg_dump**)). This is only needed, however, if you used PostgreSQL *previously*.

Now you still need to decide if you want to update using the text-oriented YaST or with the graphical YaST2. The decision is made for you if you want to update from a SuSE Linux before version 6.0 – in this case you should use YaST and proceed as described in Section 11.1.3. From version 6.0 onwards you can update using YaST2.

11.1.2 Updating with YaST2

After the preparations mentioned in Section 11.1.1 on the page before, you should first boot.

> Note
>
> You can only carry out an update with YaST2 if you want to update a SuSE Linux 6.x. For older versions (e. g.SuSE Linux 5.3) YaST1 is the right method; see Section 11.1.3.

Start YaST2 and select 'Update exisiting system' from the menu.

This module updates your current system to the latest version of SuSE Linux. The module will go through several steps. It will check, which packages need to be updated. You are able to decide this action for every single package, if you like to. Using this module you are not able to update the base system. To do this, you have to boot from the CD/DVD or the floppy disk and select 'Update exisiting system' in YaST2.

11.1.3 Updating with YaST1

Updating the base system

Because central components of the system (such as libraries) must be replaced when updating the base system, this task cannot be performed during normal operations, i.e. from the currently running Linux system.

You need to start the update environment – normally using either the supplied "boot disk" or by entering **manual** at the kernel boot prompt.

Essentially the following steps are necessary:

1. As soon as the kernel has booted from the “boot disk” or the update CD , linuxrc is started automatically.

2. In the main menu of linuxrc, in the menu ‘`Settings`’, you need to choose the language, monitor and keyboard. Press ‘`OK`’ when you are finished.

3. Now make sure that you load all the drivers you need by selecting ‘`Kernel modules`’. The exact procedure is given in the linuxrc description in Section 16 page 300.

4. After this has been done, selecting ‘`Installation / Startup system`’ and ‘`Startup installation`’ leads you to the selection of your source medium (see Section 16 page 301).

5. linuxrc then loads the installation environment and starts YaST automatically.

In YaST’s main menu, select ‘`Update existing system`’. YaST then tries to determine the root partition and displays the result. Here you can now select your root partition, as mentioned above (for example: `/dev/sda3`).

YaST then reads the existing “old” `/etc/fstab` and mounts the partitions it found in this file. Now select ‘`Continue`’.

After returning to the main menu by pressing Esc, choose ‘`Updating your system`’ (Section 3.2 page 51). Your old system is now analyzed by YaST and the results are displayed in a list.

After this, all the central configuration files of the system are updated. YaST creates backup files of those files that have changed since the last installation.

When the base system has been updated, you will find yourself in YaST’s special update mode, where you can decide on which packages to update. In addition, old configuration files will be saved as `.rpmorig` or `.rpmsave` (see Section 11.3.2 page 286). This procedure is recorded in `/var/adm/inst-log/installation-*`, and may be consulted later on.

Updating the Rest of the System

When the base system has been updated, you will then be placed in YaST’s special update mode. Here you may update the rest of your system as desired.

YaST then builds two lists. The first list shows those packages YaST has recognized, and where it has decided that an update could be useful. The second list shows you those packages where this is not so straightforward (perhaps the old package still works, but provides no information about its version).

Using these two lists, you can decide which packages should be updated. When you begin the update, all packages selected will be replaced by the new ones – all files will be saved that have been changed since the last installation.

After completing this task, you should proceed as with a first-time installation. One of the things you will need to do is select a new kernel; YaST will offer you this option.

Tip

If you use loadlin for booting you need to copy the *new* kernel and possibly your `initrd` as well to the `loadlin` directory on your DOS partition!

If you do not want the system to be rebooted in its normal runlevel (see standard runlevel Section 13.2 page 311), but would like YaST to complete the installation, you should enter the following at the LILO prompt:

```
NO_AUTO_SETUP=true
```

This parameter is especially useful if you encounter problems when booting again. These might occur if you access important parts of your system, for example via PCMCIA-SCSI. To continue with the prepared configuration, the following steps could be taken:

1. Enter the following parameter at the LILO prompt:

   ```
   NO_AUTO_SETUP=true
   ```

2. Change to user `'root'` and start YaST with: **`yast -nomenu`** to complete its configuration tasks.

3. Start **`/lib/YaST/bootsetup.conf`** as `'root'`.

Experienced Linux users might not want to start in the default runlevel, but in the "single user mode". This can be achieved by entering **`single`** at the LILO prompt.

11.1.4 Updating Single Packages

With SuSE Linux, you can update single packages whenever you want; when doing so, you must make sure that the system remains consistent: recommendations on updating can be found at: `http://www.suse.de/de/support/download/updates/`.

In YaST's package list (see Section 3.4.3 page 72), you can move around as you please. If you select a package which is needed for the system to run, you will be warned by YaST. Such packages should be replaced only in update mode. For example, many packages contain *shared libraries* which could be in use when you want to run an update. Updating shared libraries on a running system will inevitably cause problems.

11.2 From Version to Version

The following sections list details of things which have been changed from one version to the next. In this overview such things as configuration file syntax changes and abnormal behavior of well-known programs are listed. Only those anomalies are listed which might cause problems for the administrator or user.

This list is probably incomplete. Please also consult the Support Database which can be found in the package sdb, series doc (cf. Section 1.4 page 4).

Known problems and other abnormal features of each version will be announced on our webserver when found: see the links given below. Important updates of individual packages can be found at http://www.suse.de/en/support/download/updates/.

11.2.1 From 6.4 to 7.0

Problems and Special Features:
file:/usr/share/doc/sdb/de/html/bugs70.html.

- All kernel modules ("drivers") are included in the respective kernels installed (single kernel, multi-processor kernel etc.); this guarantees that the matching compiled modules are installed.
- The X Window System 4.0 no longer supports some very old graphics cards and a few very new ones; see Section 5.2 page 126. The setup program will be aware of this and switch to the preceding version 3.3.x, which is also included, accordingly.

  The development environment of version 3.3.x should also be installed if you want to build packages in accordance with the "Linux Development Platform Specification"; see Section 12.3.1 page 292.
- `ypserv` from the package `ypserv` is no longer linked to the "tcp-wrapper" library, but instead uses `/var/yp/securenets`. After an update the settings from `/etc/hosts.allow` or from `/etc/hosts.deny` should be transferred to `/var/yp/securenets`.
- The portmapper is started via `/sbin/init.d/portmap`(since 7.1 via `/etc/init.d/portmap`) or with the command **`rcportmap`**; `/sbin/init.d/rpc` is now obsolete.
- In accordance with the FHS *File system Hierarchy Standard* (see Section 12.1.1 page 291), all architecture-dependent documentation can now be found in `/usr/share/doc` (previously `/usr/doc`).
- The package `bttv` is no longer included; the kernel modules are integrated into the respective kernel installed.
- The gtk-XmHTML library is available as a separate package `gxmhtml`; it has been moved from the package `gnlibs`.
- The package `gnadmin` has been discontinued; logview is now contained in the package `gnutils`.
- On the package `cron`: in accordance with the FHS, the cron tables are located in `/var/spool/cron/tabs`; see Section 12.3.2 page 292.
- On the package `postgres`: PostgreSQL and all of its components have been reorganized on the basis of the original packages. The package `pg_datab` with the initialization database is no longer needed; if needed, initialization is performed by the startup script.

11.2.2 From 7.0 to 7.1

Problems and Special Features:
http://sdb.suse.de/sdb/de/html/bugs71.html.

- *Kernel:* The "low-level" drivers for special EIDE chip sets are integrated into the standard kernel; k_eide or eide are no longer necessary as separate images. This means you no longer need to create your own boot disk if you require this driver.
- The meaning of *runlevel* has changed; see Table 13.1 page 312. The **init** scripts are now located in /etc/init.d; if your have already created your own scripts, you should make sure they are backed up before you update them.
- **DEFAULT_LANGUAGE**: New name for the former variable **LANGUAGE** in /etc/rc.config; see Section 12.9 page 308.
- The /etc/resolv.conf is now written directly from YaST as well as YaST2; no longer by SuSEconfig.
- The limitation of the package names to 8 characters has been removed and many packages may now have more descriptive names. These new names will be taken into consideration in the update and the familiar programs will continue to be available to you automatically. If you are carrying out a re-installation, you need to be careful that the packages are selected under their new names.
- Regarding the former package ypclient: This package is now divided into ypbind and yp-tools and the init-script is called ypbind.
- Regarding package jade_dsl: In order to avoid conflict with package rzsz, the command line tool **sx** is now called **s2x** and **sgml2xml**.

11.2.3 From 7.1 to 7.2

Problems and Special Features:
http://sdb.suse.de/sdb/de/html/bugs72.html.

- package nkitb no longer exists; the programs moved into several packages: talk, rsh, finger, rwho, telnet, etc., into the -server packages. E.g. the command **ping** can now be found in package iputils.
- Renaming of several packages (see page ??); e. g.:
  series sgm: package docbook-dsssl-stylesheets, package docbook_3 and package docbook_4.
- You have to install package mod_php4 instead of package mod_php now.
- Emacs was split up into several packages:
  - Base package emacs.

  - In addition you have to install package `emacs-x11`, which includes the program *with* X11 support.
  - package `emacs-nox` contains the program *without* X11 support.
  - package `emacs-info`: Online documentation in the info format.
  - package `emacs-el` contains uncompiled library files written in Emacs Lisp – not required for a runtime environment.
  - Numerous additional packages which can be installed, if required: package `emacs-auctex` (for LaTeX); package `psgml` (for SGML/XML); package `gnuserv` (for a Client-/Server environment), etc.

- Changes to follow the current FHS (see Section 12.1.1 page 291):
  - `/media` (was `/cdrom` and `/floppy`)

  Compatible links will be available for a transitional period.

11.3 RPM—the Distribution Package Manager

In SuSE Linux, RPM (**`rpm`**), the “Red Hat Package Manager” serves as the package manager. It makes the powerful RPM database available to everyone: users, system administrators and, of course, package builders, from which detailed information on installed software can be queried at any time.

`rpm` does the following:

- compiles software applications from “pristine” sources and packages them for installation
- installs, upgrades and cleanly uninstalls software packaged in the RPM format
- supports queries, including dependencies, about packages and maintains the RPM database of installed packages

The reader is referred to the manpage for **`rpm`** (**`man rpm`**) and the book *Maximum RPM*, Bailey, 1997, Red Hat, for more information on building RPM packages. The other capabilities of RPM are briefly described below.

Installable RPM archives are packed in a special binary format. These archives consist of the program files to be installed, as well as certain meta-information which is used during the installation by **`rpm`** to configure the software package, or stored in the RPM database for documentation purposes. RPM archives normally have the extension `.rpm`.

11.3.1 Verifying Package Authenticity

Since Version 7.1, the SuSE RPM packages have a GnuPG signature:

```
1024D/9C800ACA 2000-10-19 SuSE Package Signing Key <build@suse.de>
Key fingerprint = 79C1 79B2 E1C8 20C1 890F  9994 A84E DAE8 9C80 0ACA
```

With the command

```
earth:~ # rpm --verbose --checksig apache-1.3.12.rpm
```

you can verify the signature of an RPM package and thus determine whether it really originates from SuSE or from another trustworthy facility; this is especially recommended for update packages downloaded from the Internet. Our public package signature key normally resides in `/root/.gnupg/`.

11.3.2 Managing Packages: Install, Update and Uninstall

Normally, RPM archive installation can be quickly taken care of:

```
earth: # rpm -i <package>.rpm
```

With this command, the package will be installed – but only if its dependency requirements are met and if it does not conflict with another package. With an error message, **`rpm`** requests those packages which need to be installed to fulfill dependencies. In the background the RPM database ensures that no conflicts will arise – a specific file can only belong to one package. By choosing different options you can force **`rpm`** to ignore these defaults, but be sure to know what you are doing, otherwise you risk compromising the integrity of the system and you might jeopardize the ability to update the system.

Use `-U` or `--upgrade` to update a package. This option will remove the files of the old version and immediately install the new files. **`rpm`** updates configuration files more cautiously:

- If a configuration file has *not* been changed by the system administrator, **`rpm`** will install the new version of the appropriate file. No action by the system administrator is required.
- If a configuration file was changed by the system administrator *before* the update, **`rpm`** will save the changed file with the extension `.rpmorig` or `.rpmsave` (backup file) and install the version from the new package, but only when the originally installed file and the newer version are different. If this is the case, you should compare the backup file (`.rpmorig` or `.rpmsave`) with the newly installed file and make your changes again in the new file. Afterwards, be sure to delete all `.rpmorig` and `.rpmsave` files to avoid problems with future updates.

The `-U` switch is *not* just an equivalent to un-installing with the (`-e`) option and installing with the (`-i`) option. Use `-U` whenever possible.

Note

After every update, you should check all backup files created by **`rpm`**. These are your old configuration files. If necessary, incorporate your customizations from the backup files for the new configuration files. After this process the files with the extensions `.rpmorig` and `.rpmsave` should be deleted.

To remove a package, enter the command:

```
earth: # rpm -e <package>
```

rpm will only delete the package if there are no unresolved dependencies. Theoretically it isn't possible to un-install an old `libc` using **rpm** as long as another program still needs it to work properly—the RPM database guards against this.

11.3.3 RPM Queries

With the `-q` option, **rpm** initiates queries, making it possible to inspect an RPM archive (by adding the option `-p`) and also to query the RPM database of installed packages. Several switches are available to specify the type of information required (see Table 11.1).

`-i`	Package information
`-l`	File list
`-f <FILE>`	Query a package owned by <FILE> (the full path must be specified with <FILE>)
`-s`	File list with status information (implies `-l`)
`-d`	list only documentation files (implies `-l`)
`-c`	list only configuration files (implies `-l`)
`--dump`	File list with complete details (to be used with `-l`, `-c` or `-d`)
`--provides`	List features of the package which another package can request with `--requires`
`--requires`, `-R`	Capabilities the package requires
`--scripts`	(Un-)installation scripts (pre/post install/un-install)

Table 11.1: The most important RPM query options (`-q [-p]` ... <package>)

For example, the command

```
earth: # rpm -q -i rpm
```

will display something like the following information:

```
Name        : rpm                          Relocations: (not relocateable)
Version     : 3.0.3                             Vendor: SuSE GmbH, Germany
Release     : 3                       Build Date: Wed 21 Jul 1999 13:48:52
Install date: Wed 21 Jul 1999 18:44:33 MEST      Build Host: euler.suse.de
Group       : unsorted                     Source RPM: rpm-3.0.3-3.src.rpm
Size        : 5108780                          License: GPL
Packager    : feedback@suse.de
Summary     : RPM Package Manager
Description :
RPM Package Manager is the main tool for managing software packages
of the SuSE Linux distribution.
%rpm can be used to install and remove software packages; with rpm it's easy
%to update packages.  rpm keep track of all these manipulations in a central
%database. This way it is possible to get an overview of all installed
%packages; rpm also supports database queries.
```

Option `-f` only works if you specify the complete filename with its full path. You can name as many filenames as you want: for example,

```
rpm -q -f /bin/rpm /usr/bin/wget
```

will lead to the following result:

```
rpm-3.0.3-3
wget-1.5.3-55
% rpm-2.4.1-1
% wget-1.4.5-2
```

If you know only a part of the filename, you will have to use a shell script such as the following:

```
#! /bin/sh
for i in ‘rpm -q -a -l | grep  $1 ‘; do
    echo you’ll find "\"$i\" in package":
    rpm -q -f $i
    echo ""
done
```

File contents 11.3.1: Script to search for packages

With the command

```
earth: # rpm -q --changelog rpm
```

You can have a precise list of information displayed (updates, configuration, modifications etc.) on a specific package; this example is on the package *rpm*. With the help of the RPM database, verification checks can be made. These checks are initiated with the option `-V` (or `-y`, or `--verify`). With this option, **`rpm`** will show all files in a package which have been changed since first being installed. **`rpm`** uses eight character symbols to give some hints about the kind of change (see Table 11.2):

`5`	MD5 check sum
`S`	File size
`L`	Symbolic link
`T`	Modification time
`D`	Major and minor device numbers
`U`	Owner
`G`	Group
`M`	Mode (permissions and file type)

Table 11.2: RPM verify options

In the case of configuration files, the letter `c` will be printed. For example, if you have changed `/etc/wgetrc` from the package `wget`, you may see:

```
earth: # rpm -V wget
```

```
S.5....T c /etc/wgetrc
```

The files of the RPM database are placed in `/var/lib/rpm`. If the partition `/usr` has a size of 1 GB, this database can occupy nearly 30 MB, especially after a complete update. If the database is much bigger than expected, it is useful to rebuild the database with the option `--rebuilddb`. But first make a backup of the old database. The **`cron`** script **`cron.daily`** makes daily copies of the database (packed with gzip) and stores them in `/var/adm/backup/rpmdb`. The number of copies is controlled by the variable <MAX_RPMDB_BACKUPS> (default: `5`) in `/etc/rc.config`. The size of a single backup is approximately 3 MB. (This value is valid for a 1 GB `/usr` partition.) You must take this space requirement into account when deciding how large you want the root partition to be. If `/var` has its own partition, you don't have to worry about this.

11.3.4 Installing and Compiling Source Packages

All source packages of SuSE Linux are located in the series `zq` (Source packages) and carry an `.spm` extension ("Source RPMS").

> **Tip**
>
> These packages can be handled in just the same way as all other packages. The packages, however, will not be found in the RPM database (and are not marked with an `[i]` in YaST), as only "installed" software is listed.

The directories of rpm in `/usr/src/packages` must exist (if none of your own settings have been made, for example in `/etc/rpmrc`).

`SOURCES` this is for the original sources (`.tar.gz`-files etc.) and for distribution-specific adjustments (`.dif`-files).

`SPECS` for the "spec" files, similar to a meta Makefile, which control the "build" process.

`BUILD` All the sources are unpacked, patched and compiled in this directory.

`RPMS` This is where the completed "binary" packages are stored.

`SRPMS` here are the "source"-RPMs.

> **Note**
>
> Please don't experiment with essential system packages such as package `libc`, package `rpm`, or package `nkit`, etc.! This could lead to a malfunctioning system!

When you install a source package from series `zq` with YaST , all the necessary components will be installed in `/usr/src/packages`: the sources and the adjustments in `SOURCES`, and the relevant `.spec` file in `SPECS` [1]. For our example

[1] For "making packages" see [Bai97]. Further information can be obtained from the manpage for **`rpm`** (**`man rpm`**)

we will choose the `wget.spm` package. After you have installed the package with YaST you should have the following files:

```
/usr/src/packages/SPECS/wget.spec
/usr/src/packages/SOURCES/wget-1.4.5.dif
/usr/src/packages/SOURCES/wget-1.4.5.tar.gz
```

`rpm -b <X> /usr/src/packages/SPECS/wget.spec` starts the compilation. Here <X> is a wildcard for different stages of the build process (see the output of the **`rpm -help`** or the RPM documentation). Here is a short explanation:

`-bp` Prepare sources in `/usr/src/packages/BUILD`: unpack and patch.

`-bc` the same as **`-bp`**, but with additional compilation.

`-bi` the same as **`-bp`**, but with additional installation of the built software. Caution: if the package does not support the BuildRoot feature, you might overwrite configuration files.

`-bb` the same as **`-bi`**, but with the additional creation of the "binary" package. If the compile was successful the binary should be in `/usr/src/packages/RPMS`.

`-ba` the same as **`-bb`**, but with the additional creation of the "source RPM". If the compilation was successful the binary should be in `/usr/src/packages/SRPMS`.

`-short-circuit` lets you skip specific steps.

This binary RPM may now be installed by invoking **`rpm -i`** or even better, with **`rpm -U`** (to make it appear in the RPM database).

11.3.5 Other Tools for Working with RPM Archives

The Midnight Commander (**`mc`**) is able to "browse" RPM archives and to operate on parts of them. This tool works on an RPM package archive as if the archive were a regular file system. Using mc, you can view `HEADER` information with F3 and you can copy parts of an archive with F5.

xrpm is a new graphical RPM manager, written in Python, which supports commands to FTP-accessed archives.

KDE can use the tool krpm, a graphical interface in the X Window System, for RPM management. krpm is currently in an early development stage.

Using the Alien (**`alien`**) Perl script, it is possible to convert or install an "alien" binary package. This tries to convert "old" TGZ archives to RPM before installing. This way the RPM database can keep track of such a package after it has been installed. But beware: **`alien`** is still "alpha" software, according to its author.

Last but not least, there is YaST …

12 Special Features of SuSE Linux

12.1 File System Hierarchy Standard (FHS) and the Linux Standard Base (LSB)

12.1.1 File System Hierarchy Standard (FHS)

SuSE Linux strives as far as possible to conform to the *File system Hierarchy Standard* (FHS, package `fhs`, series `doc`; see also `http://www.pathname.com/fhs/`). For this reason it is sometimes necessary to move files or directories to their "correct" places in the file system.

12.1.2 Linux Standard Base (LSB)

SuSE supports the *Linux Standard Base* project; current information on this can be found at: `http://www.linuxbase.org`.

At this time, the LSB project exists in Version 0.3; among other things, the system initialization is defined here; refer to Chapter 13 page 311.

12.1.3 *teTeX* – TEX in SuSE Linux

teTeX is put together according to the *TEX Directory Structure* (TDS) (see `ftp://ftp.dante.de/tex-archive/tds/`), without compromising the FHS.

12.2 Example Environments for FTP and HTTP

About FTP

To make it easier to set up an FTP server, the package `ftpdir` package includes an example environment. This is installed in `/usr/local/ftp`.

About HTTP

Apache is the standard web server in SuSE Linux; together with the installation of Apache some example documents are made available in `/usr/local/httpd`. If you want to set up your own web server, it is recommended that you enter your own **`DocumentRoot`** in `/etc/httpd/httpd.conf`.

12.3 Hints on Special Software Packages

12.3.1 The packages `xdevel` and `xdevel33`

The “Linux Development Platform Specification” specifies that XFree86 3.3.6 is to be used for building purposes, so that the software in question will run on as many platforms as possible. From SuSE Linux 7.0, however, XFree86 4.0 is the default. So that SuSE Linux can continue to meet the criteria of the specification, the package `xdevel33` is currently available.

If you want to build a package with SuSE Linux 7.2 in accordance with the “Linux Development Platform Specification”, you must uninstall the package `xdevel` and install the package `xdevel33` instead.

12.3.2 Package `cron`

The cron tables are now located in `/var/cron/tabs` (and not, as previously, in `/var/lib/cron`). `/etc/crontab` serves as a system wide cron table. You need to enter the name of the user who should run the command directly after the timetable (see File contents 12.3.1, here `'root'` is entered); package-specific tables, located in `/etc/cron.d` have the same format – see manpage for **`cron`** (**`man 8 cron`**).

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

File contents 12.3.1: Example of an entry in `/etc/crontab`

`/etc/crontab` cannot be processed with **`crontab -e`**, but must be loaded directly into an editor, modified, and then saved.

A number of packages install shell scripts to the directories `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` and `/etc/cron.monthly`, whose instructions are controlled by `/usr/lib/cron/run-crons`. `/usr/lib/cron/run-crons` is run every 15 minutes from the main table (`/etc/crontab`); this guarantees that processes which may possibly have been neglected can be run at the proper time. You should not be surprised if, shortly after booting the user `'nobody'` turns up in the process tables and is highly active; this probably means that `'nobody'` is just updating the locate database (see Section 13.6 page 326).

12.3.3 package **`curses`**

On the CD you will find the package `ncurses`. The corresponding libraries are named `libncurses.so.`<xx>. This means that some Makefiles have to be adapted to link **`ncurses`**. Packages of your own should be linked with the command **`-lncurses`**, and never with **`-lcurses`**. For those who want to use **`curses`** anyway, try:

```
-I/usr/include/termcap -I/usr/include/curses
-L/usr/lib/termcap -L/usr/lib/curses
```

12.3.4 Sources for `uucp`

The sources for package `uucp` are currently contained as a sub-package in the source-RPM of the program Sendmail.

12.3.5 Manpages

For some GNU applications (e. g., tar) the manpages are no longer maintained. They have been replaced by info files. Info (**`info`**) is GNU’s hypertext system. Typing **`info info`** gives you a starting help in using **`info`**. **`info`** can be launched via **`emacs -f info`** or on its own: **`info`**. The programs **`tkinfo`** and **`xinfo`** are easy to use, or you can access the help system.

12.3.6 The Command `ulimit`

With the **`ulimit`** (*user limits*) command it is possible to set limits for the use of system resources, and to have these displayed. **`ulimit`** is especially useful for limiting the memory available for applications. Through this an application can be prevented from using up too much (all) memory on its own; this could bring the system to a standstill.

`ulimit` can be used with various options. To limit memory usage, you can use the options listed in Table 12.1.

`-m`	maximum size of physical memory
`-v`	maximum size of virtual memory (swap)
`-s`	maximum size of the stack
`-c`	maximum size of the core files
`-a`	display of limits set

Table 12.1: **`ulimit`**: Setting resources for the user

System-wide settings can be made in `/etc/profile`; there, creating core files must be enabled, which is needed by programmers for “debugging”. A normal user cannot increase the values specified in `/etc/profile` by the system administrator, but he can make special entries in his own `~/.bashrc`.

```
# Limits of physical memory:
ulimit -m 98304

# Limits of virtual memory:
ulimit -v 98304
```

File contents 12.3.2: **`ulimit`**: Settings in `~/.bashrc`

Details of memory must be specified in KB.

For more detailed information take a look at manpage for **`bash`** (**`man bash`**).

12.4 Booting with the "initial ramdisk"

Describing the Problem

As soon as the Linux kernel has been booted and the root file system (`/`) mounted, programs can be run and further kernel modules can be integrated to provide additional functions.

To be able to mount the root file system in the first place, certain conditions must be met. The kernel needs the corresponding drivers to be able to access the device on which the root file system is located (especially SCSI drivers). The kernel must also contain the code which is needed in order to be able to read the file system (`ext2`, `reiserfs`, `romfs` etc.). It is also conceivable that the root file system is already encrypted; in this case a password is needed to mount the file system.

If you just take a look at the problem of SCSI drivers, then a number of different solutions are possible: the kernel could contain all imaginable drivers. This might be a problem because different drivers could conflict with each other. Also the kernel will become very large because of this. Another possibility is to provide different kernels, each one containing just one or a few SCSI drivers. This method also has the problem that a large number of different kernels are required, a problem which is then increased by the differently optimized kernels (CPU optimization, SMP).

The idea of loading the SCSI driver as a module leads to the general question which is answered by the concept of an *initial ramdisk*: creating a way of being able to perform userspace-programs even before the root file system is mounted.

12.4.1 Concept of the Initial Ramdisk

The *initial ramdisk* (also called "initdisk" or "initrd") solves precisely the problems described above. The Linux kernel provides an option of having a small file system loaded to a RAM disk and running programs there before the actual root file system is mounted. The loading of `initrd` taken over by the boot loader (LILO, loadlin etc.); all these boot loaders only need BIOS routines to load data from the boot medium. If the boot loader is able to load the kernel, then it can also load the initial ramdisk. Special drivers are not required.

12.4.2 The Order of the Booting Process with initrd

The boot loader loads the kernel and the `initrd` to memory and starts the kernel. The boot loader informs the kernel that an `initrd` exists and where it is located in memory.

If the `initrd` was compressed (which is typically the case), then the kernel de-compresses the `initrd` and mounts it as a temporary root file system. A program called linuxrc is started on this in the initrd. This program can now do all the things necessary to be able to mount the proper root file system. As soon as linuxrc finishes, the (temporary) initrd is unmounted and the boot process continues as normal, with the mounting of the proper root file system. Mounting the initrd and running linuxrc can thus be seen as a short interlude during a normal boot process.

If the initrd cannot be unmounted (which is normally regarded as an error), the kernel will try and re-mount initrd to the directory /initrd. If the mount point /initrd does not exist, an error message will be displayed. In such a case the system is fully functional, but the memory occupied by initrd cannot be released at all and is no longer available.

linuxrc

These are the only requirements for the program linuxrc in the initrd; it must have the special name linuxrc and it must be located in the root directory of the initrd. Apart from this, it only needs to be executable by the kernel. This means that linuxrc may be dynamically linked; in this case, the "shared libraries" in /lib must be completely available in initrd. linuxrc can also be a shell script. For this to work, a Shell must exist in /bin. In short, initrd must contain a minimal Linux system, which allows the program linuxrc to be carried out. When SuSE Linux is installed, a statically linked linuxrc is used, to be able to keep initrd as small as possible (space on boot disks is very limited). linuxrc is carried out with 'root' permissions.

12.4.3 Boot Loaders

Most boot loaders (above all, LILO, loadlin and syslinux) can handle initrd. Individual boot loaders are given instructions on how to use initrd as follows:

1. LILO

Enter the following line in /etc/lilo.conf:

```
initrd=/boot/initdisk.gz
```

The file /boot/initdisk.gz is the *initial ramdisk*. It can be, but doesn't have to be, compressed.

2. loadlin.exe

call up with:

```
C:> loadlin <kernelimage> initrd=C:\loadlin\initdisk.gz <parameter>
```

3. syslinux

Enter the following line in syslinux.cfg:

```
append initrd=initdisk.gz <further parameters>
```

12.4.4 Using `initrd` in SuSE

Installing the System

The `initrd` has already been used for some time for the installation: here the user can load modules and make the entries necessary for an installation (above all, for the source medium). Linuxrc then starts YaST, which carries out the installation. When YaST has finished, it tells linuxrc where the root file system of the freshly installed system is located. linuxrc writes this value to `/proc`, terminates, and informes the kernel to continue booting into the newly installed system.

For an installation of SuSE Linux you are thus, from the very beginning, booting the system which is just being installed – somehow, quite clever. A real re-boot after installation only takes place if the kernel does not match with the modules which were installed in the system. Since SuSE Linux only uses a kernel for uni-processor systems, this will only happen if an SMP kernel was installed in the system with the corresponding modules. For this reason, an SMP kernel newly installed in the system must be rebooted, in order for it to be able to use all modules.

Booting the Installed System

In the past, YaST has provided more than 40 kernels for installing in the system, whereby the only basic difference in the kernels was that each of them contained a specific SCSI driver. This was necessary to be able to mount the root file system after booting. Further drivers could then be loaded afterwards as modules.

Because optimized kernels are now available, this concept is no longer feasible – by now, over 100 kernel-images would be needed.

This is why an `initrd` is used now, even to start the system normally. The way it is used is similar to that for an installation. The linuxrc used here, however, is simply a shell script which just has the task of loading a given module. Typically, this is just one single module, namely the very SCSI driver which is needed to access the root file system.

Creating an `initrd`

An `initrd` is created by means of the script **`mk_initrd`**. In SuSE Linux, the modules to be loaded are specified by the variable **`INITRD_MODULES`** in `/etc/rc.config`. After installation this variable is automatically occupied by the correct values (the installation linuxrc knows which modules were loaded). Here it should be mentioned that the modules were loaded in exactly the same order in which they appear in **`INITRD_MODULES`**. This is especially important if a number of SCSI drivers are used, since otherwise the names of the hard disks would change. Strictly speaking, it would be sufficient just to load those drivers needed to access the root file system, because the automatic loading of additional SCSI drivers may cause problems (how should it be “triggered”, if hard disks hang on the second SCSI adapter), we load all SCSI drivers needed at the installation, by means of `initrd`.

The current **`mk_initrd`** checks if a SCSI driver is needed for the root file system. If you run **`mk_initrd`** on a system where `/` is to be found on EIDE hard disks, an `initrd` is not needed as the kernel used for SuSE Linux already contains the EIDE driver. As there are more and more special EIDE controllers coming onto the market, in the future it will probably be necessary in such cases to use an `initrd` to boot the installed system.

> Note
>
> Because the loading of the `initrd` with the boot loader runs in just the same way as loading the kernel itself (LILO notices in its `map` file the location of the files), LILO must be re-installed after every change in `initrd`! Thus after every **`mk_initrd`**, a **`lilo`** is also necessary.

12.4.5 Possible Difficulties – Self-Compiled Kernels

A self–compiled kernel can often lead to the following problems: out of habit the SCSI driver is hard-linked to the kernel, but the existing `initrd` remains unchanged. When you boot, the following occurs: the kernel already contains the SCSI driver, the hardware is detected. `initrd` however now tries to load the driver again as a module; with some SCSI drivers (especially with the `aic7xxx`), this leads to the system blocking. Strictly speaking, this is a kernel error (an already existing driver should not be allowed to be loaded again as a module). The problem is already known from another context, however (serial drivers).

There are several solutions to the problem: either configure the driver as a module (then it will be correctly loaded in the `initrd`) or remove the entry for `initrd` from the bootloaders configuration file (`/etc/lilo.conf`) . An equivalent to the latter solution is to remove the variable **`INITRD_MODULES`** and then run **`mk_initrd`**, which then realizes that no `initrd` is needed.

12.4.6 Prospects

It is quite possible in the future that an `initrd` will be used for many more and much more sophisticated things than loading modules needed for access to `/`.

- “High end” EIDE drivers
- Root file system on RAID software (linuxrc sets up the `md` devices)
- Root file system on the LVM
- Root file system is encrypted (linuxrc asks for the password)
- Root file system on a SCSI hard disk on a PCMCIA adapter.

Further Information

`/usr/src/linux/Documentation/ramdisk.txt`
`/usr/src/linux/Documentation/initrd.txt`
manpage for **`initrd`** (**`man 4 initrd`**).

12.5 linuxrc

linuxrc is started during the boot up of the kernel, usually as a prelude to a Linux system installation, before the "real" booting commences (the kernel must first be properly configured, of course). This allows you to boot a small, modularized kernel and to load the few drivers you may need as modules, from a floppy disk – in an emergency even from a second disk (`modules`).

linuxrc is your assistant for loading all relevant hardware drivers. You can also use linuxrc as a boot disk for an already installed system, for example, as a rescue disk. You can even start a totally independent RAM disk based rescue system, for example, if something serious should happen to your hard disk or you have simply forgotten your `'root'` password. More about this below, in Section 12.6 page 302.

Main Menu

After you have selected the language, screen and keyboard, you find yourself in linuxrc's main menu. Start an installation with '`Start installation / system`'. Whether you can go there directly depends mainly on your hardware.

If all components that are needed for installation have already been recognized by the kernel, you do not need to load additional drivers. This mainly applies to machines that only have (E)IDE adapters (and only (E)IDE hard disks and CD-ROM as well).

If there is a SCSI adapter installed which is needed for installation,[1] you have to load the corresponding SCSI module. The same applies if you want to install via an existing network. Here, the appropriate module has to be loaded first.

Furthermore, there are a lot of older CD-ROMs that are driven by proprietary controllers and which, therefore, need their own kernel modules. If PCMCIA devices are connected to a laptop, you need these modules as well.

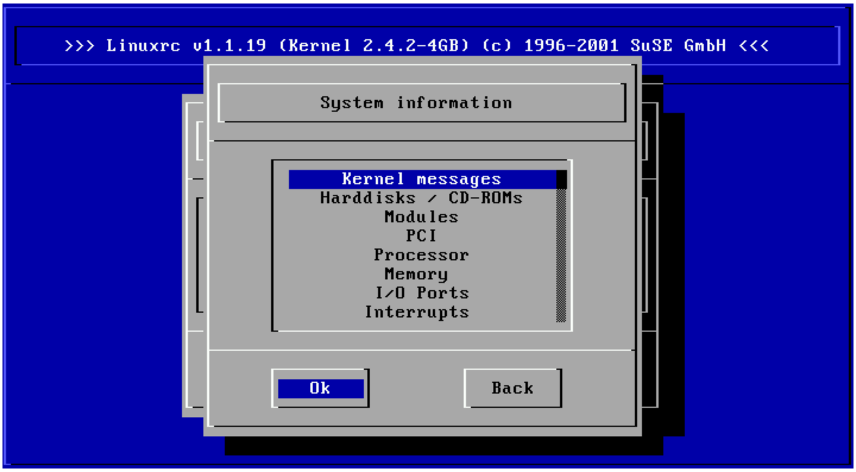
System Information

If you are not sure about your hardware, the boot messages might help you.

In '`System information`' (Figure 12.1 on the next page) you can check a number of other things, besides kernel messages, such as the I/O-addresses of PCI cards or the size of the main memory recognized by Linux.

[1] An adapter with only a scanner connected to it is not required at boot time.

You can check some system information in '`System information`'. Here, you can check the used interrupts, I/O ports used, main memory and recognized PCI devices as detected by Linux.



```
>>> Linuxrc v1.1.19 (Kernel 2.4.2-4GB) (c) 1996-2001 SuSE GmbH <<<
```

Figure 12.1: System information

The next lines show how a hard disk and a CD-ROM connected to an (E)IDE controller announce their start. In this case, you do not need to load additional modules:

```
hda: ST32140A, 2015MB w/128kB Cache, LBA, CHS=1023/64/63
hdb: CD-ROM CDR-S1G, ATAPI CD-ROM drive
Partition check:
 hda: hda1 hda2 hda3 < hda5 >
```

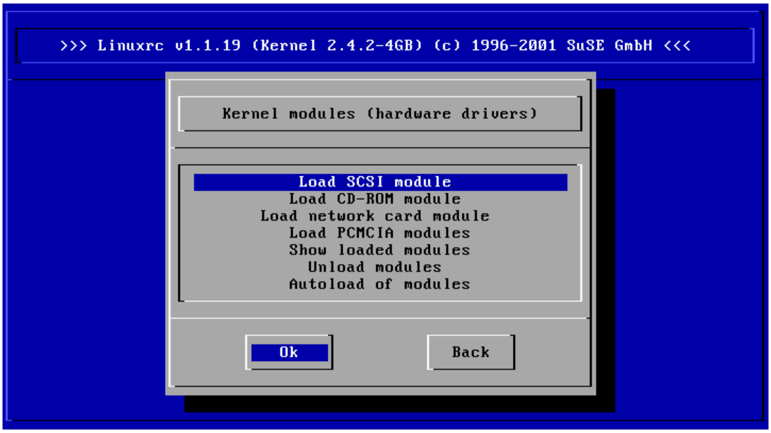
If you booted a kernel that already has a SCSI driver compiled, you do not need this SCSI driver as a module as well. Quite typical announcements when loading SCSI adapters and connected devices might look like this:

```
scsi : 1 host.
Started kswapd v 1.4.2.2
scsi0 : target 0 accepting period 100ns offset 8 10.00MHz FAST SCSI-II
scsi0 : setting target 0 to period 100ns offset 8 10.00MHz FAST SCSI-II
  Vendor: QUANTUM   Model: VP32210           Rev: 81H8
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 0, lun 0
scsi0 : target 2 accepting period 236ns offset 8 4.23MHz synchronous SCSI
scsi0 : setting target 2 to period 248ns offset 8 4.03MHz synchronous SCSI
  Vendor: TOSHIBA   Model: CD-ROM XM-3401TA  Rev: 0283
  Type:   CD-ROM                             ANSI SCSI revision: 02
scsi : detected 1 SCSI disk total.
SCSI device sda: hdwr sector= 512 bytes. Sectors= 4308352 [2103 MB] [2.1 GB]
Partition check:
 sda: sda1 sda2 sda3 sda4 < sda5 sda6 sda7 sda8 >
```

Loading Modules

You select which kinds of modules you need. If you booted via disk, the corresponding data has to be read by linuxrc and displayed in a list.

If you have booted from CD or from DOS (via loadlin), these modules are already set in linuxrc. This saves tedious loading, but it needs additional memory.

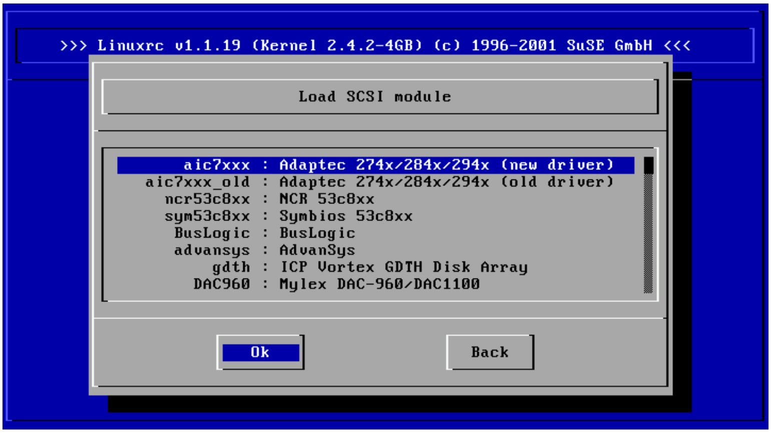


```
>>> Linuxrc v1.1.19 (Kernel 2.4.2-4GB) (c) 1996-2001 SuSE GmbH <<<
```

Figure 12.2: Load modules

linuxrc offers you a list of available drivers. On the left, there is the name of the module and, on the right, a short description of its usage.

For some components, there are a variety of drivers to choose from (even newer alpha-code drivers).



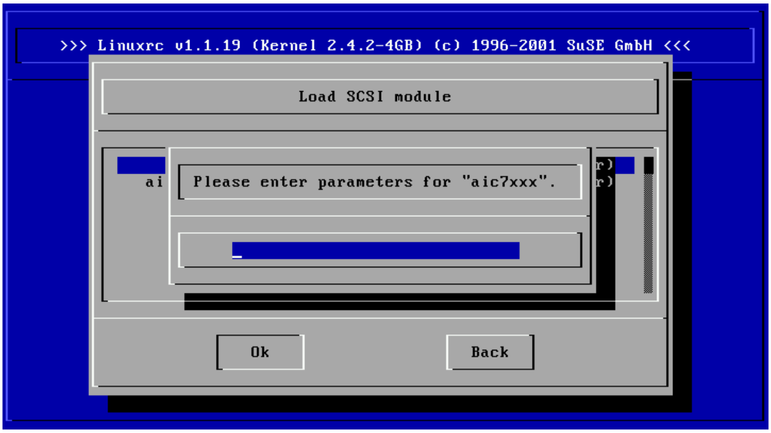
```
>>> Linuxrc v1.1.19 (Kernel 2.4.2-4GB) (c) 1996-2001 SuSE GmbH <<<
```

```
      aic7xxx : Adaptec 274x/284x/294x (new driver)
  aic7xxx_old : Adaptec 274x/284x/294x (old driver)
    ncr53c8xx : NCR 53c8xx
    sym53c8xx : Symbios 53c8xx
     BusLogic : BusLogic
     advansys : AdvanSys
         gdth : ICP Vortex GDTH Disk Array
       DAC960 : Mylex DAC-960/DAC1100
```

Figure 12.3: Selection of SCSI drivers

Passing on Parameters

When you have found a suitable driver, move to it with the cursor and press ⏎ . Now there is a dialog box where you can add additional parameters for this module.



```
>>> Linuxrc v1.1.19 (Kernel 2.4.2-4GB) (c) 1996-2001 SuSE GmbH <<<
```

Load SCSI module

Please enter parameters for "aic7xxx".

Figure 12.4: Entering parameters for a module to be loaded

We would like to point out that , in contrast to the LILO prompt, parameters for the same module have to be separated by blanks.

In most cases, it is not necessary to specify the hardware in detail. Most drivers find their components automatically. Most network cards and proprietary CD-ROM drives, however, need parameters. If in doubt, just try ⏎.

Recognizing and initializing certain hardware can take some time. Switching to virtual console #4 (Alt + F4) lets you watch the kernel messages while loading. SCSI drivers need a while, as they have to wait for each device to load.

The messages are displayed by linuxrc so you can verify that everything ran smoothly. If it fails, the messages might give you a hint as to why it failed.

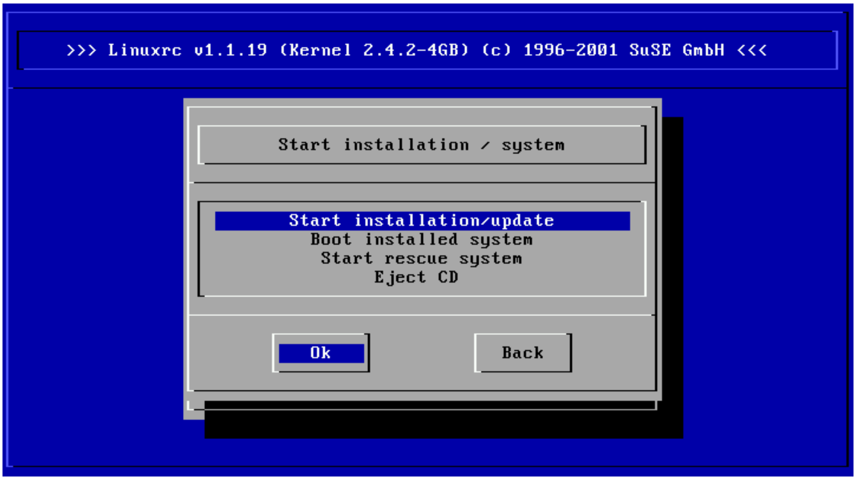
Start Installation / System

Once you have set up hardware support via modules, you can switch to the 'Start installation / system' menu.

From here (Figure 12.5 on the following page) a number of procedures can be started: 'Start installation' (from this item an update is also started), 'Boot installed system' (the root partition must be known), 'Start rescue system' (refer to Section 12.6 on the next page) and 'Eject CD'.

'Start Live-CD' is only available if you booted a "LiveEval-CD" You may download ISO images from the ftp server (live-eval-<VERSION>):

ftp://ftp.suse.com/pub/suse/i386/



```
>>> Linuxrc v1.1.19 (Kernel 2.4.2-4GB) (c) 1996-2001 SuSE GmbH <<<
```

Figure 12.5: the linuxrc '`Start`' menu

Tip

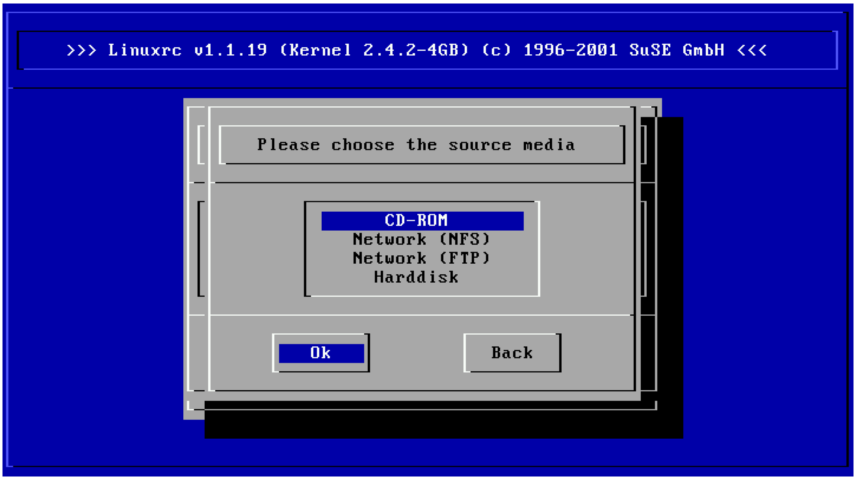
The item '`Start live-CD`' can be of great use if, for example, you want to test, without actually installing, if the computer in question or the Notebook you might want to buy, is at all compatible with SuSE Linux – such a test ought to be possible in every modern PC shop, without any trouble!

For the installation (Figure 12.6 on the facing page) you can choose various sources for the installation and similarly for the rescue system (see Figure 12.6 on the next page).

12.6 The SuSE Rescue System

SuSE Linux contains several Linux rescue systems. With their help, you can reach your Linux partitions on the hard disks in emergencies again "from the outside": a boot disk or the "Rescue" System, that you can load from disk, CD, the network, or the SuSE FTP server. There is also a bootable SuSE Linux CD (the *"LiveEval-CD"*) which can be used a rescue system.

Furthermore, the rescue system includes several help programs with which you can remedy large problems with inaccessible hard disks, misconfigured configuration files, etc.



```
>>> Linuxrc v1.1.19 (Kernel 2.4.2-4GB) (c) 1996-2001 SuSE GmbH <<<
```

Please choose the source media

Figure 12.6: Selection of source media in linuxrc

Tip

Always invest some of your time in a boot and rescue disk, since the slight effort for creating and maintaining the disk is negligible in comparison to the time wasted and work involved when, in the case of emergency, you cannot access your system or your CD-ROM drive.

12.6.1 Preparations

For setting up your rescue system, you need two disks free of errors: one as future boot disk and the other for the compressed image of a small root file system. The image file `bootdisk` for booting the system and the file `rescue` for the Root file system can be found on the first CD inthe `/disks` directory.

There are three ways to set up the root file system on the disk:

- with YaST
- in a console using Linux commands

```
earth: # /sbin/badblocks -v /dev/fd0 1440
earth: # dd if=/media/cdrom/disks/rescue of=/dev/fd0 bs=18k
```

- at the DOS prompt (where the CD-ROM drive is `Q:`)

```
Q:\> cd \dosutils\rawrite
Q:\dosutils\rawrite> rawrite.exe
```

The rescue disk is based on, as of Version 7.2, libc5 (SuSE Linux 5.3), since it is possible in this SuSE Linux version to save some programs such as Editor, fdisk or e2fsck to a disk.

Note

The rescue disk cannot be mounted, since it is not a file system: But only contains the compressed images of one. If you wish to read the file system, then read the following paragraph.

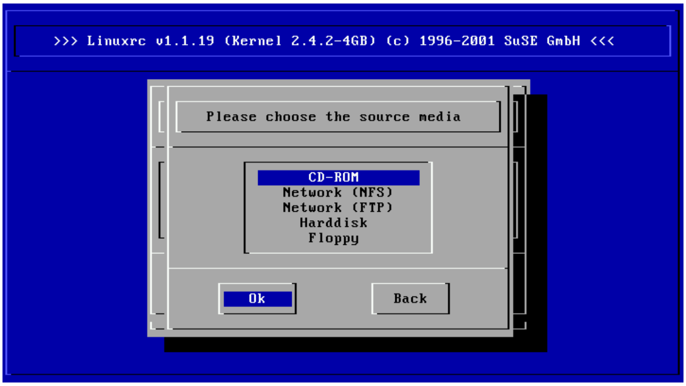
If you want to read the image, you will need to de compress the image file and then mount the de compressed image as user `'root'`. If your Linux kernel support the *loop device*, the procedure is as follows:

```
earth: # cp /media/cdrom/disks/rescue /root/rescue.gz
earth: # gunzip /root/rescue.gz
earth: # mount -t ext2 -o loop /root/rescue /mnt
```

12.6.2 Starting the Rescue System

The rescue system is launched using the SuSE boot disk or from your boot-able SuSE Linux CD 1. It is required that the disk and CD-ROM drives are bootable; if necessary, you will need to change the boot series in the CMOS set-up. Following are the steps for starting the rescue system:

1. Start your system with the SuSE boot disk and with the first SuSE-Linux CD inserted in your CD-ROM drive.
2. You can launch the entire system or at the boot prompt either select '`Rescue`' or '`Manual`', where you can define which kernel modules should be loaded.
3. Make the respective settings for language, keyboard and screen.
4. Select the item '`Installation/Start system`' in the main menu.
5. If you started with the boot disk, you should now insert the installation-CD or the `rescue` disk with the compressed image of the rescue system.



```
>>> Linuxrc v1.1.19 (Kernel 2.4.2-4GB) (c) 1996-2001 SuSE GmbH <<<
```

Please choose the source media

Figure 12.7: Source medium for the `rescue` system

6. In the menu '`Start installation/system`' select the item '`Start rescue system`' and then specify the desired source medium (Figure 12.7 on the facing page, page 304).
   Subsequently, we will introduce a few tips on selection options:

   '`CD-ROM`': When loading the rescue system, the path `/cdrom` is exported. This makes the Installation from *this* CD possible.

   > **Note**
   >
   > You now still need to enter the required values in SuSEconfig (Section 13.5 page 316 pp.).

   '`Network (NFS)`': To start the `rescue` system via NFS from the network, you have to have the driver for your network card already installed;

   '`Network (FTP)`': To start the `rescue` system via FTP from the network, you have to have your network card driver ready!

   '`hard disk`': Load the `rescue` system from the hard disk.

   '`Floppy Disk`': The `rescue` system can also be started from the floppy disk; especially if the computer only has a small amount of working memory.

Regardless of which medium you have chosen, the rescue system will be decompressed, loaded onto a RAM disk as a new root file system, mounted and started. Now it is ready for use.

12.6.3 Working with the Rescue System

The rescue system provides three virtual consoles on keys `Alt` + `F1` to `Alt` + `F3`. Here '`root`' may log in without a password. `Alt` + `F4` brings you to the system console where you can view the kernel and syslog messages.

A shell and lots of other useful utilities (net tools), such as the mount program, can be found in the `/bin` directory. In `sbin` you can find important file and network utilities for reviewing and repairing the file system (e.g. e2fsck).

Furthermore, this directory contains the most important binaries for system maintenance, such as fdisk, mkfs, mkswap, mount, mount, init, shutdown, as well as ifconfig, route, and netstat for maintaining the network.

An editor, vi, is located in `/usr/bin`. Also, tools like grep, find, less, etc., along with telnet are available.

Accessing Your Normal System

To mount your SuSE Linux system using the rescue system, you should use the mount point `/mnt`. You can also use or create another directory.

As an example, let's assume that your normal system is put together according to the `/etc/fstab` shown in the example file 12.6.1 on the following page.

```
/dev/sdb5       swap         swap         defaults   0   0
/dev/sdb3       /            ext2         defaults   1   1
/dev/sdb6       /usr         ext2         defaults   1   2
```

File contents 12.6.1: Example `/etc/fstab`

> **Caution**
>
> Pay attention to the order of steps outlined in the following section for mounting the various devices.

To be able to access you entire system, mount it step by step in the `/mnt` directory using the following commands:

```
earth:/ # mount /dev/sdb3 /mnt
  earth:/ # mount /dev/sdb6 /mnt/usr
```

Now you can access your entire system and, e. g., correct mistakes in configuration files such as `/etc/fstab`, `/etc/passwd`, and `/etc/inittab`. The configuration files are now located in the `/mnt/etc` directory, instead of in `/etc`.

To recover even completely lost partitions with the fdisk program by simply setting it up again, determine where on the hard disk the partitions were previously located, make a hardcopy print-out of the `/etc/fstab` directory as well as the output of the command

```
earth: # fdisk -l /dev/<disk>
```

Instead of the <disk> variable, you should insert, in order, the device names of your hard disks, i.e. `hda`.

Repairing File Systems Damaged file systems are tricky problems for the rescue system. This could happen after an unscheduled shutdown caused by power failure or a system crash. Generally, file systems cannot be repaired on a running system. If you encounter really severe problems, you may not even be able to mount your root file system and have the system boot end up in a `"kernel panic"`. Here, the only chance is to repair the system from the “outside” using a rescue system.

The SuSE Linux rescue system contains the utilities e2fsck and dumpe2fs (for diagnosis). These should remedy most problems.

In an emergency, manpages often are not available. That is why we have included them in this manual in Appendix B page 331.

Example:

If mounting a file system fails due to an *invalid* superblock, then the e2fsck program would most probably fail too. If this were the case, your superblock may be corrupted too. There are copies of the superblock located every 8192 blocks (8193, 16385, . . .). If your superblock got corrupted, you can try one of the copies instead. This is accomplished by entering the command:

```
earth: # e2fsck -f -b 8193 /dev/damaged_partition
```

The **-f** option forces the file system check and overrides e2fsck's error so that – since the superblock copy is intact – everything is fine.

12.7 Virtual Consoles

Linux is a multi-user and multi-tasking system. You will learn to appreciate the advantages of these features, even on a standalone PC system.

In text mode, there are six virtual consoles available. You can switch between them by using Alt +F1 to Alt + F6. The seventh console is reserved for X11.[2]

If you want to switch to a console from X11 without leaving X11, you should use Ctrl + Alt + F1 to Ctrl + Alt + F6. Alt + F7 brings you back to X11.

12.8 Keyboard Layout

In order to standardize the keyboard mapping of programs, changes were made to the following files:

```
/usr/X11R6/lib/X11/Xmodmap
/etc/inputrc
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/lib/X11/app-defaults/XTerm
/usr/share/emacs/20.7/site-lisp/term/*.el
/usr/lib/joerc
```

These changes only affect applications that make use of **terminfo** entries or whose configuration files are changed directly (**vi**, **less**, etc.). Other non-SuSE applications should be adjusted to these defaults.

[2] You can assign more consoles via /etc/inittab.

12.9 Local Adjustments

SuSE Linux is to a very large extent internationalized and can be adjusted flexibly to local events; in other words, internationalization ("I18N") allows specific localizations ("L10N"). The abbreviations I18N and L10N are derived from the first and last letters of the words, and in between, the number of letters omitted.

Settings are made via **LC_** variables, which are defined in the file `/etc/rc.config`. This is less to do with "native language support", and more to do with the categories for News (Language), *Character Set*, *Sort Order*, *Time and Date*, *Numbers* and *Money*. Each of these categories can be defined directly via its own variable, or indirectly via an overlying variable in the file `rc.config` (see manpage for **locale** (**man 5 locale**)):

1. **RC_LC_MESSAGES**, **RC_LC_CTYPE**, **RC_LC_COLLATE**, **RC_LC_TIME**, **RC_LC_NUMERIC**, **RC_LC_MONETARY**: These variables are passed on to the shell without the **RC_** prefix and determine the above categories; the files concerned are listed below. **DEFAULT_LANGUAGE** is *not* exported.

   The current setting can be shown with the command **locale**.

2. **RC_LC_ALL**: This variable overwrites the values of the variables mentioned in item 1, if they have been set.

3. **RC_LANG**: If none of the above variables are set, this is the "fallback". By default SuSE Linux only sets **RC_LANG**; this makes it easier for the user to enter his own values.

4. **ROOT_USES_LANG**: A `yes`/`no` variable. If it is set to `no`, then `'root'` always works in the POSIX environment.

5. **DEFAULT_LANGUAGE**: This variable has two functions. It defines the language in which YaST is started, and at the same time is the fallback if **RC_LANG** is not set. It is set by YaST, depending on the language chosen in the language selection menu. In order to avoid any misunderstanding: **DEFAULT_LANGUAGE** is not passed on to the shell by SuSEconfig as **LANGUAGE**, but the value of this variable is only used to set **LANG**, if it is required.

DEFAULT_LANGUAGE can be influenced by YaST via the item '`Select language`'; see Section 3.3.1 page 53. The other variables can be set via the `rc.config` editor; see Section 3.6.14 page 92.

The value of such a variable is put together from the "language code", "country code", "encoding" and "modifier". Individual details are connected by special characters:

```
LANG=<language>[[_<COUNTRY>].Encoding[@Modifier]]
```

Concrete Examples:

```
LANG=de_DE.ISO-8859-15
```

This is how the German language is defined with the character set `ISO-8859-15`; this character set contains the Euro character, but the default currency here is DM.

And an example for setting an option:

```
LANG=no_NO@nynorsk
```

You should also use the language and country codes together. Language settings follow the standard, ISO 639 (http://www.indigo.ie/egt/standards/iso639/iso639-1-en.html), country codes are listed in ISO 3166 (http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html. This only makes sense, however, if just those values are chosen for which usable description files can be found in /usr/lib/locale. Further description files can be created, using the command **localedef**, from the files in /usr/share/i18n. A description file for de_DE@euro.UTF-8 is thus created with:

```
earth:~ # localedef -i de_DE@euro -f UTF-8 de_DE@euro.UTF-8
```

SuSEconfig reads the variables in /etc/rc.config and writes the necessary changes to /etc/SuSEconfig/profile and /etc/SuSEconfig/csh.cshrc. /etc/SuSEconfig/profile is read by /etc/profile and /etc/SuSEconfig/csh.cshrc by /etc/csh.cshrc. This makes the settings available system-wide.

The users can override the system settings by editing ~/.bashrc. So, if the system settings are set to en_US, the user can e. g.switch to german system messages:

```
LC_MESSAGES=de_DE
```

12.9.1 Settings for Language Support

It should be pointed out that the files in the category *Messages* are as a rule only stored in the language directory (e. g.de) in order to have a fallback. If you set **LANG** to de_AT and the "message" file in /usr/share/locale/de_AT/LC_MESSAGES does not exist, it will fall back to /usr/share/locale/de/LC_MESSAGES.

A fallback chain can also be defined; e. g.for Breton → French or for Galician → Spanish → Portuguese:

```
LC_MESSAGES="br_FR:fr_FR"
LC_MESSAGES="gl_ES:es_ES:pt_PT"
```

Possible Problems

- The thousand point is not recognized. **LANG** is probably set to de. Since the description which the glibc uses is located in /usr/share/locale/de_DE/LC_NUMERIC, **LC_NUMERIC** for example must be set to de_DE.

Further Information:

- *English-Howto* file:/usr/share/doc/howto/en/html/English-HOWTO.html

- Markus Kuhn, *UTF-8 and Unicode FAQ for Unix/Linux*, currently at: `http://www.cl.cam.ac.uk/~mgk25/unicode.html`.
- *Unicode-Howto* by Bruno Haible `file:/usr/share/doc/howto/en/html/Unicode-HOWTO.html`.

13 The SuSE Linux Boot Concept

Booting and initializing a UNIX system can challenge even an experienced system administrator. This chapter gives you a short overview of the SuSE Linux boot concept. The implementation at hand transcribes the *System Initialization* section of the LSB outline (Version 0.3); refer to Section 12.1.2 page 291 for more information on LSB.

The simple words `"Uncompressing Linux..."` signal that the kernel is taking control over your hardware. It checks and sets your console – or more precisely: the BIOS registers of graphics cards and output format – to read BIOS settings and to initialize basic hardware interfaces. Next, your drivers "probe" existing hardware and initialize it accordingly. After checking the partitions and mounting the root file system, the kernel starts init which "boots up" (Unix jargon) the main system with all its programs and configurations. The kernel will control the entire system, including hardware access and the CPU time programs may use.

13.1 The init Program

The program init is responsible for correctly initializing all system processes. Thus, it is the father of all processes in the entire system.

init takes a special role: init is started directly by the kernel and resists *signal 9*, which normally enables you to kill processes. All further programs are either started directly by init or by one of its "child" processes.

init is centrally configured via the `/etc/inittab` file. Here, the "runlevels" are defined (more about runlevels in the next section). It also specifies which services and daemons are available in each of the levels.

Depending on the entries in `/etc/inittab`, several scripts are invoked by init which, for reasons of clarity, all reside in the directory `/etc/init.d`.

The entire process of starting up the system (and shutting down, as well) is maintained by init. From this point of view, the kernel can be considered a "background process" whose task it is to maintain all other processes and to adjust CPU time and hardware access according to requests from other programs.

13.2 Runlevels

In Linux, *runlevels* define how the system is going to be started up. After booting, the system starts as defined in `/etc/inittab` in the line **`initdefault`**.

Usually this is 3 or 5(see Table 13.1).An alternative to this is assigning a special runlevel at boot time (e. g., at the LILO prompt) : the kernel passes any parameters which it doesn't need directly to init.

To change runlevels while the system is running, you can just enter init with the appropriate number. Only the super user is allowed to do this:

```
root@earth:/ > init 1
```

brings you to *single user mode* which is used for the maintenance and administration of your system. After finishing work in *S* mode, the system administrator can change the runlevel to 3 again by typing:

```
root@earth:/ > init 3
```

Now all essential programs are started and users can log in and work with the system. Table 13.1 below gives you an overview of available runlevels. Runlevel 2 should not be used on a system whose /usr partition is mounted via NFS:

Runlevel	Meaning
0	System halt
S	Single user mode; from boot prompt with US keyboard layout
1	Single user mode
2	Local multi-user without remote network (standard)
3	Full multi-user with network
4	Unused
5	Full multi-user mode with network and xdm
6	System reboot

Table 13.1: Valid runlevels in Linux

You can halt the system using:

```
root@earth:/ > init 0
```

or cause it to reboot using:

```
root@earth:/ > init 6
```

If you have already installed and configured the X Window System properly (Section 5 page 125) and want users to log in via a graphical user interface, you can easily change the runlevel to 5. You should give it a try first by typing:

```
root@earth:/ > init 5
```

to see whether the system works as you expected.

Tip

With a damaged /etc/inittab, you can end up in a system which cannot be brought up properly. Therefore, be extremely careful while editing /etc/inittab! To repair the damage, you may try to enter **init=/bin/sh** at the LILO boot prompt for directly booting into a shell. This looks like: **boot: linux init=/bin/sh**

13.3 Changing Runlevels

Generally, there are a couple of things that happen if you change runlevels. First, *stop scripts* of the current runlevel are launched, closing down some programs which are essential for the current runlevel. Thereafter, *start scripts* of the new runlevel are started. Here, in most cases, a number of programs will be started.

To illustrate this, we will show you a change from runlevel 3 to 5:

- The administrator (`'root'`) tells init to change runlevels:

  ```
  root@earth:/ > init 5
  ```

- init now consults its configuration file (`/etc/inittab`) and realizes that it should start `/etc/init.d/rc` with the new runlevel as a parameter.

- Now rc calls all the stop scripts of the current runlevel, but only for those where there is no start script in the selected new runlevel. In our example, these are all the scripts which reside in `/sbin/init.d/rc3.d` (old runlevel was 3) and which start with a `'K'`. The number following `'K'` guarantees a certain order to start, as there are some dependencies which have to be taken into consideration.

  > **Note**
  >
  > The names of the stop scripts always begin with `'K'` for kill, and those for the start scripts with `'S'` for start.

- The last thing to start are the start scripts of the new runlevel. These are (in our example) in `/sbin/init.d/rc5.d` and begin with an `'S'`. The same procedure regarding the order in which they are started is applied here.

If you want to change to the same runlevel which you are already in, init only checks `/etc/inittab` for changes and starts the appropriate steps (e. g., for starting a getty on another interface).

13.4 Init Scripts

Scripts in `/etc/init.d` are divided into two sections:

- scripts which are executed directly by init. This only applies while booting and shutting down the system immediately (power failure or pressing `Ctrl` + `Alt` + `Del` by the user).

- scripts which are started indirectly by init. This happens when the runlevel is changed. Generally, /etc/init.d/rc is executed here, which guarantees the correct order of the relevant scripts.

All scripts are located in `/etc/init.d`. Scripts for changing the run level are also found there but are called via symbolic links from one of the subdirectories `/etc/init.d/rc0.d` to `/etc/init.d/rc6.d`. This is just for clarity reasons and avoids duplicate scripts (e. g., if they are used in several runlevels). Since every script can be executed as both a start and a stop script, these scripts have to "understand" the parameters "start" and "stop". The scripts understand, in addition, the **`restart`**, **`reload`**, **`force-reload`** und **`status`** options; what these different options mean is elaborated in Table 13.2.

Option	Meaning
`start`	Starts service
`stop`	Stops service
`restart`	Stops service and restarts if service is already running; otherwise starts the service over again
`reload`	Read in configuration of service again without stoppping and restarting it
`force-reload`	Read in configuration of the service again if the support supports this; in this way different than **`restart`**
`status`	show current status

Table 13.2: Overview of init script options

After leaving runlevel 3, `/etc/init.d/rc3.d/K40network` will be called up; `/etc/init.d/rc` calls up the `/etc/init.d/network` script by means of the **`stop`** parameter. After entering runlevel 5, the same script will be started, this time, however, with the **`start`** parameter.

Links in these runlevel-specific subdirectories simply serve to allocate the scripts to a certain runlevel.

Adding and removing the required links is done by the program insserv (or by the link `/usr/lib/lsb/install_initd`) when installing and uninstalling the package; refer to manpage for **`insserv`** (**`man 8 insserv`**)

Below, we give you a short introduction to the boot and stop scripts that are launched first (or last, respectively) as well as an explanation of the maintaining script.

- *boot*
  Executed while starting the system directly using init. It is independent of the chosen runlevel and is only executed once. Here, file systems are checked, the kernel daemon is launched, some unnecessary files in `/var/lock` are deleted and the network is configured for the loopback device (if it has been selected in `/etc/rc.config`). Furthermore, the system time is set up with the `isapnp` tools and Plug and Play hardware configured (refer to Section 6.2 page 174). .

  If an error occurs while automatically checking and repairing the file system, the system administrator can intervene after first entering the root password.

The directory `/etc/init.d/boot.d` is assigned to this script. Every script, whose name starts with an `‘S’` in this directory is executed while the system is booted. This is the right place for your own personal extensions, which should be executed only once.

Last to be executed is the script boot.local.

- *boot.local*
  Here, you can enter additional commands to be executed at boot time before changing into a runlevel. It can be compared to `AUTOEXEC.BAT` on DOS systems.

- *boot.setup*
  General settings that have to be performed while changing from *single user mode* to another runlevel.

  Here, keyboard maps are loaded and the kernel daemon is started, which loads modules automatically.

- *halt*
  This script is only executed while changing into runlevel 0 or 6. Here, it is executed either as **`halt`** or as **`reboot`**. Whether the system shuts down or reboots depends on how halt is called.

- **rc**
  This script is of overriding importance whenever runlevels are changed. It calls the appropriate stop scripts of the current runlevel and the start scripts of the newly selected runlevel.

With this concept in mind, you can create your own scripts. A skeleton has been prepared in `/etc/init.d/skeleton`. The exact format is described in the LSB outline; this defines specifically the order of steps and in which levels the script has to be processed.

You may add your own scripts to this skeleton very easily. A template may be found in `/etc/init.d/skeleton`. To enable a script via `/etc/rc.config`, it is recommended that you create a <START_> variable in this file. Additional parameters should only be added if really needed (refer to the `/etc/init.d/gpm` script for reference).

Now you need to create the links in the corresponding `rc?.d` to your script to make sure it is launched when you change runlevels (see above Section 13.3 page 313 for script names, etc.). The manpage for **`init.d`** (**`man 7 init.d`**) gives you all the necessary technical background.

Caution

Please handle these scripts with the utmost care! A faulty script may hang your machine! See Section 13.2 page 311 if all else fails. . .

13.5 /etc/rc.config and SuSEconfig

Practically the entire configuration of SuSE Linux can be done via a central configuration file called `/etc/rc.config`. Here, a couple of environment variables are set which are (among others) checked by the init scripts. Each of the scripts in `/etc/init.d` executes `/etc/rc.config` as a first step, in order to read the values of those variables which apply to it.

> **Note**
>
> From SuSE Linux 6.0 packages with extensive adjustment options store their variables in individual files in the directory `/etc/rc.config.d`; this is the case, for example, with the package `sendmail` or the package `i4l` (ISDN).
> To simplify things, however, we will usually just mention `/etc/rc.config`, even if the file has been moved somewhere else!

Moreover, a large number of configuration files can be generated from `/etc/rc.config`. This is the task of /sbin/SuSEconfig. If you change the network configuration, for example, the file `/etc/host.conf` will be regenerated, as it is dependent on the configuration you have made.

So, if you change `/etc/rc.config` manually, you should invoke /sbin/SuSEconfig afterwards to make sure all changes to the appropriate configuration files are made at the correct places. If you change the configuration with YaST, you don't have to bother. YaST automatically executes /sbin/SuSEconfig and updates your configuration files.

This concept enables you to make basic changes to your configuration without having to reboot the system. Since some changes are rather complex, some programs probably have to be restarted in order for the changes to take effect. This procedure is explained more fully in network configuration, where these programs are forced to be restarted using the commands:

```
earth:~ # rcnetwork stop
earth:~ # rcnetwork start
```

As you can see, you can easily start and stop init scripts by hand.

Generally, we recommend the following steps for configuring your system:

- Bring the system into *single user mode* (Runlevel 1):

  ```
  earth:~ # init 1
  ```

- Change the configuration file `/etc/rc.config` as needed. This can be done using an editor of your choice or with YaST in '`Changing configuration file`' (see Section 13.6 on the facing page).

- Execute /sbin/SuSEconfig to make the changes take effect. If you have changed /etc/rc.config via YaST, this is done automatically.

- Bring your system back to the previous runlevel:

  ```
  root@earth:~ > init 3
  ```

This procedure is mainly relevant if you have changed system-wide settings (such as network configuration). It is not necessary to go into *single user mode* for small changes, but it ensures that all relevant programs are correctly restarted.

Tip

To *categorically* disable the automatic configuration of SuSEconfig you need to set the variable <ENABLE_SUSECONFIG> in `/etc/rc.config` (please note Section 13.6). By using selected `rc.config` variables, you may partially disable the auto configuration.

13.6 The Variables in `/etc/rc.config`

In this section, we describe all the parameters of the system, including their default settings. If you don't use YaST to change `/etc/rc.config`, make sure you set "empty" parameters as two quotation marks (e. g., **`KEYTABLE=""`**) and that you surround parameters that contain a blank with quotation marks (parameters consisting of only one word do not have to be quoted). In our description, each parameter is given a value in order to make its settings as clear as possible:

Basic Settings

- **`ENABLE_SUSECONFIG=yes`**
  With this entry you can disable SuSEconfig completely. Please don't contact our Installation Support if you have trouble configuring your system after disabling SuSEconfig ;-)

- **`MAIL_REPORTS_TO=newbie`**
  SuSEconfig can mail reports (created by YaST or included in packages) to you. Here you can set the address. If you don't want this feature, simply set it to **`""`**.

- **`GMT=-u`**
  If your hardware is set to GMT (*Greenwich Mean Time*), you should set this variable to `-u`.[1] Otherwise, leave it empty. This setting is relevant for automatic switching to summer or winter time.

- **`TIMEZONE=CET`**
  The time zone where you live and where you operate your computer. This setting is also relevant for automatic switching to summer or winter time.

[1] `-u` is an abbreviation for *universal time*.

- **DEFAULT_LANGUAGE="english"**
  Set by YaST when changes made using 'Define language' setting option (refer to Section 3.3.1 page 53). This value can also be classified as fallback if **RC_LANG** is not set; in this case, the value entered in /usr/share/locale/locale.alias will be resolved.

- **RC_LANG="en_US"**
  Sets LANG for locale; this way, a default can be set for the local users. This value is comes in to play as long as no specific **RC_LC_*** variables are used. The relevant rc.config variables are as follow: **RC_LC_ALL** (This can overwrite LC_* as well as LANG!), **RC_LC_MESSAGES**, **RC_LC_CTYPE**, **RC_LC_MONETARY**, **RC_LC_NUMERIC**, **RC_LC_TIME** und **RC_LC_COLLATE**. Refer to Section 12.9 page 308.

- **ROOT_USES_LANG="no"**
  Should local settings also be used for 'root'?

- **ENABLE_SYSRQ="no"**
  View inside of the kernel. Before implementing it, definitely read /usr/src/linux/Documentation/sysrq.txt!

Initializing Local Hardware (keyboard, modem, mouse,PCMCIA, etc.)

- **KEYTABLE=de-latin1-nodeadkeys**
  Defines keymaps.

- **KBD_NUMLOCK=no**
  NumLock on/off.

- **KBD_CAPSLOCK=no**
  CapsLock on/off.

- **KBD_TTY="tty1 tty2 tty3 tty4 tty5 tty6"**
  NumLock and CapsLock can be limited to certain TTYs; "" stands for all TTYs.

- **KBD_RATE=30**
  Sets the automatic keyboard frequency. Possible values are from twice to 30 times per second. For this change to take effect, keyboard delay has to be set as well.

- **KBD_DELAY=250**
  This is the delay whereafter automatic repetition of the pressed key commences. This value is in milliseconds but isn't very accurate. You have to assign **KBD_RATE** as well.

- **CONSOLE_FONT=mr.fnt**
  This is the console font. Not all fonts support German umlauts. YaST provides a little window where you can test all fonts and choose the one you like best. Additional settings are: **CONSOLE_SCREENMAP**, **CONSOLE_UNICODEMAP** and **CONSOLE_MAGIC**.

- **MODEM=/dev/ttyS1**
  Interface to which the modem is connected. YaST and SuSEconfig, in turn, create a link from /dev/modem to the device specified.

- **MOUSE=/dev/ttyS2**
  Interface to which the mouse is connected. YaST and SuSEconfig, in turn, create a link from /dev/mouse to the device specified.

- **START_GPM=yes**
  Set to yes to start mouse console support. This enables you to exchange text between consoles using the mouse. gpm can cause problems in connection with certain bus mice. If you encounter problems while starting X, you should disable gpm. The other alternative is to start xdm, since gpm is not started in runlevel 3.

- **GPM_PARAM=" -t logi -m /dev/mouse"**
  Initialization parameters for gpm. These are normally set via YaST.

- **PCMCIA=i82365**
  This is for assigning the chipset: valid entries are: i82365 and tcic. If the variable is set to **""** the PCMCIA sub-system is not launched. Fine tuning is achieved via **PCMCIA_PCIC_OPTS** and **PCMCIA_CORE_OPTS**

- **START_ISAPNP=yes**
  Initialize ISA P'n'P at boot time.

- **INITRD_MODULES="aic7xxx"**
  The names of the modules which need to be loaded when the kernel is being booted, for example, to access the hard disk. Useful or necessary entries can be made during the installation or the update; see Section 12.4 page 294.

- **START_KERNELD=yes**
  This variable determines whether the kernel daemon should be started automatically at boot time. This daemon is responsible for automatically loading kernel modules on demand. For the current standard kernel 2.2.xx, kerneld is *no longer* required. A short description of the module concept and functions of kerneld can be found in chapter Section 9.2 page 239.

Starting and Configuring Local Network and Other Services

- **START_LOOPBACK=yes**
  Sets up a sort of “mini” network created by configuring the *Loopback* device. Since many programs rely on this functionality, it should be set.[2]

- **CHECK_ETC_HOSTS=yes**
  SuSEconfig can do some checks and modifications to /etc/hosts.

- **BEAUTIFY_ETC_HOSTS=no**
  If /etc/hosts is to be sorted.

[2] Your kernel needs to have been compiled with network support for this to work.

- **SETUPDUMMYDEV=no**
  Determines whether the dummy network device should be set up: this is useful for non-permanent network connections (e. g., SLIP or PPP). If you have an ethernet card, it can be troublesome at times.

- **CREATE_HOSTCONF=yes**
  SuSEconfig can create and check `/etc/host.conf`.

- **NETCONFIG=_0**
  Specifies how many network cards (or other net devices) are installed. The text shows an example for one network card (they start with 0). For a system with two cards installed, it should resemble **NETCONFIG="_0 _1"**. For a system without networking, it should not be set.

- **IPADDR_0=193.141.17.202**
  IP address of the first networking card.

- **NETDEV_0=eth0**
  Name of the first network device (normally an Ethernet card, therefore, the example, `eth0`). Other possible settings are `str1` or `plip1`. If there is more than one card installed, additional cards are supplied with the variables NETDEV_1 to NETDEV_3.

- **IFCONFIG_0="193.141.17.205 broadcast 193.141.17.255 netmask 255.255.255.192"**
  Configuration command for the first networking device installed. These settings can easily be assigned using YaST. If you have more than one card installed, just enter the corresponding values in the appropriate variables.

- **CLOSE_CONNECTIONS=false**
  If this variable is set to `true` *and* the system runs in “runlevel” 0 or 6, **/etc/init.d/route** sends a **SIGTERM** to all processes that own an open “remote tcp” or “udp” connection.

- **IP_DYNIP=no**
  Switch on the “dynamic IP patch” when booting; if `yes`, the script `/etc/init.d/boot` enables this patch via an entry in the /proc file system.

- **IP_TCP_SYNCOOKIES=yes**
  Activate “syn flood protection”.

- **IP_FORWARD=no**
  If the machine should perform forwarding with two network cards, **IP_FORWARD** should be set to `yes`; normally this is preferable, or required, for a “router” or for “masquerading”. The script `/etc/init.d/boot` enables “IP forwarding” by means of an entry in the `/proc` file system.

- **FQHOSTNAME=earth.cosmos.com**
  Fully qualified hostname of your machine; fully qualified means the complete name, consisting of computer name and domain name.

- **`SEARCHLIST=cosmos.com`**
  This entry is used for completing a not fully qualified hostname. If, for example, you enter `venus`, a check is made on whether `venus.cosmos.com` is a valid address. This variable *has* to be set if you plan to use DNS! You should at least enter your domain name here. You can make up to three entries, which should be separated by blanks.

- **`NAMESERVER=193.141.17.193`**
  Address of the name server which is to be interrogated if a hostname has to be transposed to an IP address. You can enter up to three name servers which should be separated by blanks. If you plan to use a name server, **`SEARCHLIST`** *has* to be set!

- **`ORGANIZATION="Gladstone Ganter Inc."`**
  This text appears in every newsposting you send.

- **`NNTPSERVER=helios`**
  Address of your news server. If you receive your news via UUCP and it is saved locally, you should enter `localhost`.

- **`IRCSERVER=helios`**
  This is the place for your IRC server (*Internet Relay Chat*). Names of the servers should be separated by blanks.

- **`START_INETD=yes`**
  Controls whether the inetd super daemon should be activated. This daemon reacts to calls from other hosts and starts (depending on the port) the appropriate service. You need it if you want to log in via telnet or rlogin. If you plan to use xinetd (see Section 19), set this to `no`.

- **`START_XINETD=no`**
  Controls whether the xinetd super daemon should be activated (this is an enhanced inetd, see Section 19). If you plan to use this daemon, **`START_INETD`** should be set to `no`.

- **`START_POSTFIX=no`**
  Activates the mail server postfix. The relevant variables are
  `POSTFIX_CREATECF`, **`POSTFIX_RELAYHOST`**,
  `POSTFIX_MASQUERADE_DOMAIN` and **`POSTFIX_LOCALDOMAINS`**.

- **`SMTP=no`**
  Set to `yes` if a sendmail daemon should be activated. If you receive your e-mail exclusively via UUCP, you don't need it, provided you call sendmail -q after each polling. (rmail, started by UUCP, just puts mail into a queue but doesn't deliver it). If mail spool directories are mounted via NFS (on a network, for example) and the single host has got only outgoing mail, this could be set to *no* as well. The same applies with the use of *relay hosts*.

- **`START_PORTMAP=no`**
  Determines whether to start the portmapper or not. You need the portmapper if you plan to use your host as an NFS server. Without this daemon, rpc.mountd and rpc.nfsd can't run; this is why the portmapper is started

when the variable is set to no, but **NFS_SERVER** is switched on! It is also necessary for NIS

- **NFS_SERVER=no**
  If the host is going to be used as an NFS server, this variable has to be set to yes. This initializes the start of rpc.nfsd and rpc.mountd.

- **REEXPORT_NFS=no**
  Set this variable to yes in order to re-export mounted NFS directories or NetWare volumes.

- **NFS_SERVER_UGID=yes**
  If the daemon (rcp.ugidd) for the conversion of user and group ID's is to be started; this variable only takes effect if <NFS_SERVER> is set to yes.

- **USE_KERNEL_NFSD="no"**
  If the package knfsd is installed, the kernel-based NFS daemon can be used. With this NFS daemon, locking is possible. See the corresponding variable, **USE_KERNEL_NFSD_NUMBER**.

- **START_AUTOFS=no**
  This daemon enables you to automatically mount directories, via NFS as well as local directories, CD-ROM drives, floppy disks and more. The package autofs, series n must be installed and configured.

- **START_RWHOD=no**
  Controls whether rwhod is started. Caution: the rwhod regularly sends "Broadcasts". If you use "on-demand" connection (ISDN or diald), this may cause connections to be made, resulting in extra traffic and costs!

- **START_ROUTED=no**
  The route daemon is only necessary for dynamic routes (see the manpage for **routed** (**man routed**)). Caution, this service establishes a connection every 30 seconds. If the machine is connected to the Internet via dialup (for example, ISDN), then it makes *no* sense to set this variable to yes.

- **START_NSCD=yes**
  NSCD Initializing "Name Service Caching Daemon" at boot time.

- **START_NAMED=no**
  This controls whether the name daemon should be activated.

- **CREATE_YP_CONF=yes**
  Set to yes if SuSEconfig should create the necessary files for YP. This depends on the next two entries as well. SuSEconfig also makes the appropriate changes to /etc/passwd and /etc/group.

- **YP_DOMAINNAME=cosmos.com**
  YP domain name of hosts.

- **YP_SERVER=helios.cosmos.com**
  Name of the NIS server.

- **USE_NIS_FOR_RESOLVING=no**
  Use NIS for resolving host names.

- **START_CIPED=no**
  Start CIPE daemon for an IPIP tunnel.

- **START_DHCPD=no**
  Start server for DHCP *Dynamic Host Configuration Protocol*. For this the variables **DHCPD_INTERFACE**, **START_DHCRELAY** and **DHCRELAY_SERVERS** are used.

- **START_LDAP=no**
  Start LDAP server.

- **START_RADIUSD=yes**
  Start radius accounting and authentication service. This service is used by some ISPs for authentication of their users. See documentation in `/usr/share/doc/packages/radiusd`.

- **START_LPD=yes**
  Start the lpd ("line printer"). Normally needed for printing.

- **START_NNTPD=yes**
  Start nntpd if you want to access news via an NNTP site.

- **START_INN=no**
  Start the INN news server.

- **START_ATD=yes**
  Controls whether the AT daemon is activated. This daemon enables you to perform tasks periodically. In contrast to the cron daemon, this action is only performed once.

- **START_HTTPD=yes**
  Controls whether the Apache http daemon should be activated.

- **START_HTTPSD=yes**
  Specifies whether the Apache httpsd ("secure" webserver) with SSL and PHP3 should be started.

- **START_SQUID=no**
  Controls whether the proxy server squid should be activated.

- **DOC_HOST=""**
  If you want to use the central documentation server which contains the SuSE help system, enter the name of the host, e. g. `"helios.cosmos.com"`.

- **DOC_SERVER=no**
  You should set this variable to `yes` on the documentation server. In **DOC_ALLOW** (see below) access to http-rman is set. Furthermore the index files for the http server have been rearranged: `http://‘hostname-f‘` instead of `http://localhost`.

- **DOC_ALLOW="LOCAL"**
  List of machines (as patterns for /etc/hosts.allow) that are allowed to access the documentation server. This variable is read only if **DOC_SERVER** is set to yes. You could also enter a subdomain here (e. g. with ".cosmos.com").

- **HTTP_PROXY=""**
  A couple of programs (e. g. lynx, arena, or wget) are capable of using proxy sites if this environment variable has been set.
  SuSEconfig may set this in /etc/SuSEconfig/* (see SDB file:/usr/share/doc/sdb/de/html/lynx_proxy.html). Example: "http://proxy.provider.de:3128/".

- **FTP_PROXY=""**
  FTP proxy. Example: "http://proxy.provider.de:3128/".

- **GOPHER_PROXY=""**
  Gopher proxy. Example: "http://proxy.provider.de:3128/".

- **NO_PROXY=""**
  This enables you to exclude (sub) domains from the proxy.
  Example: "www.me.de, do.main, localhost".

- **START_HYLAFAX=no**
  Activates Hylafax. You will have to invoke faxsetup before setting this variable to yes.

- **START_SMB=no**
  Start the samba server; a file and printer server for MS-Windows.

- **START_MARSNWE=no**
  Activates the Novell server emulation.

- **START_SSHD=yes**
  Start the “Secure Shell Daemon”; ensure before starting that a “host key” exists – see also the documentation in /usr/share/doc/packages/ssh as well as the manpages.

- **START_XNTPD=yes**
  Controls whether the “Network Time Protocol (NTP) daemon” is activated (package xntp). It is configured via /etc/ntp.conf. Radio Controlled Clocks have addresses in the form 127.127.T.U; in which T is the type of clock and U is the “unit number” which has a range from 0 to 3. Most clocks use a serial port or a special bus. The device file (Device) is normally only to the symbolic link /dev/device-U which refers to the actual hardware, where U has the value of the “unit number”. See also /usr/share/doc/packages/xntp/html/refclock.htm. *Example:* You have a radio controlled clock attached to a serial port, you’ll also need the necessary Symlink. For this see refclock.htm. for the typical DCF77 receiver the “PARSE” driver is required:

  ```
  ## Type 8 Generic Reference Driver (PARSE)
  ## Address:     127.127.8.u
  ## Serial Port: /dev/refclock-u
  ```

So, if you choose, by an entry in `ntp.conf`, the value `server 127.127.8.0` you'll also need the Symlink `/dev/refclock-0` to refer to `ttySx` – where `x` represents the serial port on which the Radio controlled clock is attached.

- **`XNTPD_INITIAL_NTPDATE=""`**
  List of the NTP servers from which the time can be obtained before the local server is started; e. g."`helios.cosmos.com`". If a number of servers are entered, they must be separated by a space. (package `xntp`). It is configured via `/etc/ntp.conf`.

- **`DISPLAYMANAGER=""`**
  Sets up the login of the machine. This may either be a text console or the X Window System. Possible entries are: `xdm` (The standard display manager of the X Window System), `kdm` (KDE's display manager) or `""`. The latter sets the login to text console (runlevel 2). This is the default.

- **`KDM_SHUTDOWN=root`**
  Controls which user is allowed to shutdown the machine via kdm (reboot or shutdown). Possible values are: `root` (`'root'`), `all` (every user), `none`, and `local` (it may only be shutdown by users that logged in locally). If this is set to `""`, `root` is the default.

- **`CONSOLE_SHUTDOWN=reboot`**
  Controls how **`init`** should react to `Ctrl` + `Alt` + `Del` . Possible values: `reboot` (the machine reboots), `halt` (the machine shuts down) and `ignore` (nothing happens). Default is `reboot`.

- **`START_AXNET=no`**
  Applixware server for**Applixware** .

- **`START_MYSQL=no`**
  Server for MySQL.

- **`START_ADABAS=no`**
  Adabas server. The following variables belong to Adabas: **`DBROOT`**, **`DBNAME`**, **`DBUSER`** and **`DBCONTROL`** – see the respective comments in `rc.config`.

- **`START_DB2=no`**
  Server for DB2.

- **`START_ARKEIA=no`**
  Arkeia backup server.

- **`START_ARGUS=no`**
  Argus server (network monitor).

- **`ARGUS_INTERFACE=eth0`**
  The interface Argus should listen to.

- **`ARGUS_LOGFILE="/var/log/argus.log"`**
  The Argus logfile. This file might become quite large!

- **`CRON=yes`**
  Sets the start and stop of the *cron daemon*. This daemon lets you start certain programs at a given time. It is only started in runlevels 2 and 3. It is highly recommended that you activate this daemon, especially if your computer is running all the time. An alternative or replacement is the AT daemon (see Section 19 page 323).

> **Note**
>
> There are a lot of options which require you to regularly run certain commands and programs. Therefore, the cron daemon should be activated on every system.

Local Maintenance

- **`RUN_UPDATEDB=yes`**
  Set this to `yes` to have the locate database updated once per day via cron. locate is useful for quickly finding files. This tool may be fine tuned by a set of variables: **`RUN_UPDATEDB_AS`**, **`UPDATEDB_NETPATHS`**, **`UPDATEDB_NETUSER`**, and **`UPDATEDB_PRUNEPATHS`** (see comments in `rc.config`). – This updating may be carried out shortly after you have booted, if the computer is *not* switched on all the time; see Section 12.3.2 page 292.

- **`REINIT_MANDB=yes`**
  If the manpage data base should be renewed once a day by cron.daily.

- **`CREATE_INFO_DIR=yes`**
  Set this to `yes` to have `/usr/info/dir` created, which serves as a general index of all info pages. This is useful after installing a package which contains info pages. Keep in mind that perl needs to be installed for this to work.

- **`CHECK_PERMISSIONS=set`**
  Controls check of file permissions according to `/etc/permissions`. **`set`** corrects wrong entries, **`warn`** warns you, and **`no`** deactivates this feature.

- **`PERMISSION_SECURITY="easy local"`**
  There are three levels of security: `/etc/permissions.paranoid`, `/etc/permissions.secure` and `/etc/permissions.easy`. You may enter either **`easy`**, **`secure`** or **`paranoid`**. You can also create your own security levels; e. g. in `/etc/permissions.local`. Then you may use **`local`** to activate your level.

- **`RPMDB_BACKUP_DIR=/var/adm/backup/rpmdb`**
  Controls where cron.daily should install its RPM database backup. If you want to disable this feature, set this variable to **`""`**.

- **`MAX_RPMDB_BACKUPS=5`**
  Maximum number of backup files for the RPM database.

- **DELETE_OLD_CORE=yes**
  Core files are memory images of programs which have been killed due to a segmentation fault. These images are very useful for debugging purposes. If this is enabled, a regular search and deletion of old core files is launched.

- **MAX_DAYS_FOR_CORE=7**
  Assigns how old core files should be before they are deleted.

- **MAX_DAYS_FOR_LOG_FILES=365**
  If a log file (mainly in /var/log) reaches a certain size, it will be automatically compressed and archived. 'root' is informed via e-mail. This parameter sets how long these files are to be kept on disk before being deleted automatically. If you set this value to 0, no compressing and no archiving will be done. Log files are written continuously and can grow to a considerable size! Compressed log files can be viewed anytime using less.

- **MAX_DAYS_IN_TMP=30**
  Selected directories (see **TMP_DIRS_TO_CLEAR** below) are checked daily to see whether they have been accessed during the selected time interval (set in days). Files which have not been modified will be deleted automatically.

- **TMP_DIRS_TO_CLEAR="/tmp /var/tmp"**
  Enter here all directories which are to be searched for old files
  (see **MAX_DAYS_IN_TMP=30** above).

- **OWNER_TO_KEEP_IN_TMP="root bs"**
  Files of system users given here should not be deleted, even if they have not been modified during the time specified.

- **CWD_IN_ROOT_PATH=no**
  The current working current working directory in the path of user 'root'.

- **ROOT_LOGIN_REMOTE=no**
  If a login for 'root' is to be allowed by telnet or rlogin, this variable should be set to yes. For security aspects concerning this variable, please refer to page ??.

- **SUSEWM_UPDATE**=yes
  Controls whether SuSEconfig should adapt the system-wide configuration files according to the installed packages. You may fine tune this feature by means of the following variables: **SUSEWM_WM**, **SUSEWM_MWM**, **SUSEWM_XPM**, **SUSEWM_ADD** and **SUSEWM_COMPAT**.

A Support

In case problems should arise during the installation, you will find more information on the web page:

`http://support.suse.de/en/`

and

`http://support.suse.de/de/`

There you will find our Support Database which provides many options for analysis and problem solving. This web page also includes corrections to the handbook and additional information which could not be included at print time.

You can also find help and hints on the individual adjustment of your system (Internet, printer, sound card, etc.) in the SuSE documentation (technical handbook, package documentation) located in the directory `/usr/share/doc/packages` of your SuSE Linux system.

The previously mentioned Internet site is among some of the most frequently visited Linux help sites and is a good starting point if you have difficulties. You can also reach the free support by:

> Note
>
> Since your access to support is determined by your valid registration code, we urge you to register for each SuSE Linux version you have, whether you are upgrading or you have received the product from us directly.
> **You will not automatically be registered upon purchasing SuSE Linux.** We kindly ask you to understand that we can only offer support to registered customers and that registration extensions (for the same registration code) are not possible.

	USA	**UK**
WWW:	`http://www.suse.com`	`http://www.suse.de/en/`
E-mail:	support@suse.com	support@suse.co.uk
Phone:	+1 (510) 628-33 85	+44 845 025-0010
Fax:	+1 (510) 628-33 81	+44 20 8387-4010
Postal address:	SuSE, Inc. 580 Second Street, Suite 210 USA-Oakland, California 94607	SuSE Linux Ltd. The Kinetic Centre Theobald Street Borehamwood, WD6 4JP United Kingdom

Germany

WWW:	`http://www.suse.de`
E-mail:	support@suse.de
Phone:	+49 (421) 526-2300
Fax:	+49 (421) 526-2350
Postal address:	SuSE GmbH, Support Schanzäckerstr. 10 D-90443 Nürnberg

Please also have a look at the URLs

`http://www.suse.com/us/support/isupport/index.html`

and

`http://support.suse.de/de/services/support.html`

for details on the opening hours.

If you want to send an e-mail request directly to installation support, always specify your registration code which is located on the back of your CD package.

Installation support will help you install a SuSE Linux System and get it up and running. This also applies to central system components which assist in the basic operation. This includes:

- Installing the SuSE Linux base system from CD or DVD.
- Basic X11 graphical user interface configuration
- Basic Internet dial-in configuration with a modem, an ISDN card or by T-DSL/ADSL (client-side only).
- Configuring a printer connected directly to the computer and basic configuration of the printer service for the respective printer and machine.
- Basic configuration of a supported sound card.

Those who have the personal edition are entitled to 60 days of installation support following registration. Ninety days support is included with the Professional Edition.

B Manual Page of e2fsck

```
E2FSCK(8)                                               E2FSCK(8)

NAME
       e2fsck - check a Linux second extended file system

SYNOPSIS
       e2fsck  [  -pacnyrdfvstFSV ] [ -b superblock ] [ -B block-
       size ] [ -l|-L bad_blocks_file ][ -C fd ] device

DESCRIPTION
       e2fsck is used to check a Linux second extended file
       system.

       device is  the  special  file  corresponding to the device
              (e.g /dev/hdXX).

OPTIONS
       -a     This option does the same thing as the  -p  option.
              It is provided for backwards compatibility only; it
              is suggested that people use -p option whenever
              possible.

       -b superblock
              Instead  of  using  the  normal superblock, use an
              alternative superblock specified by superblock.
              This  option  is  normally  used  when  the primary
              superblock has  been  corrupted;  most file systems
              have  primary  superblocks  located at blocks 8193,
              16385, etc.  If an alternative superblock is speci-
              fied  and  the file system is not opened read-only,
              e2fsck will make sure that the  primary  superblock
              is  updated  appropriately  upon  completion of the
              file system check.



       -B blocksize
              Normally, e2fsck will search for the superblock  at
              various different block sizes in an attempt to find
              the appropriate block size.   This  search  can  be
              fooled in some cases.  This option forces e2fsck to
              only try locating the superblock  at  a  particular
              blocksize.   If the superblock is not found, e2fsck
              will terminate with a fatal error.

       -c     This option causes e2fsck to run  the  badblocks(8)
              program  to  find  any  blocks which are bad on the
              file system, and then marks them as  bad  by adding
              them  to  the  bad block  inode. This option causes
              e2fsck  to  write  completion  information  to  the
              specified  file  descriptor  so that  the  progress
              of the file system  check  can be  monitored.  This
              option is  typically used  by  programs  which  are
              running e2fsck. If the file descriptor specified is
              0, e2fsck will print a completion  bar  as  it goes
              about its business.  This  requires that  e2fsck is
              running on a video console or terminal.

       -d     Print  debugging  output  (useless  unless  you are
              debugging e2fsck).

       -f     Force checking even if the file system seems clean.

       -F     Flush the file system device's buffer caches before
              beginning.  Only really  useful  for  doing  e2fsck
              time trials.

       -l filename
              Add  the  blocks  listed  in  the file specified by
              filename to the list of bad blocks. The format  of
              this  file  is the same as the one generated by the
              badblocks(8) program.

       -L filename
              Set the bad blocks list to be the  list  of  blocks
              specified by filename.  (This option is the same as
              the -l  option,  except  the  bad  blocks  list  is
              cleared  before  the  blocks listed in the file are
              added to the bad blocks list.)

       -n     Open the file system readonly, and assume an answer
              of  'No'  to  all questions.  Allows e2fsck to be
              used non-interactively.  (Note: if the -c,  -l,  or
              -L  options  are  specified  in  addition to the -n
              option, then the file system will be  opened  read-
              write, to permit the bad-blocks list to be updated.
              However, no other  changes  will  be  made  to  the
              file system.)

       -p     Automatically  repair  ("preen")  the  file  system
              without any questions.

       -r     This option does nothing at  all;  it  is  provided
              only for backwards compatibility.

              -s  This  option  will byte-swap the file system so
              that it is using  the  normalized,  standard  byte-
              order  (which  is  i386  or little endian).  If the
              file system is already in the  standard byte-order,
              e2fsck will take no action.

       -S     This option will byte-swap the file system, regard-
              less of its current byte-order.

       -t     Print timing statistics for e2fsck.  If this option
              is  used  twice,  additional  timing statistics are
              printed on a pass by pass basis.

       -v     Verbose mode.

       -V     Print version information and exit.
       -y     Assume an  answer  of  ‘‘yes’’  to  all  questions;
              allows e2fsck to be used non-interactively.

EXIT CODE
       The exit code returned by e2fsck is the sum of the
       following conditions:
            0    - No errors
            1    - File system errors corrected
            2    - File system errors corrected, system should
                   be rebooted if file system was mounted
            4    - File system errors left uncorrected
            8    - Operational error
            16   - Usage or syntax error
            128  - Shared library error

SIGNALS
       The  following signals have the following effect when sent
       to e2fsck.

       SIGUSR1
              This signal causes e2fsck  to  start  displaying  a
              completion bar.  (See discussion of the -C option.)

       SIGUSR2
              This signal causes e2fsck to stop displaying a com-
              pletion bar.

REPORTING BUGS
       Almost any piece of software  will have bugs.  If  you
       manage to find a file system which causes e2fsck to crash,
       or which  e2fsck  is unable to repair, please report it to
       the author.

       Please include as much information as possible in your bug
       report.  Ideally, include a  complete  transcript  of  the
       e2fsck  run,  so I can see exactly what error messages are
       displayed. If you have a writeable file system  where  the
       transcript can be stored, the script(1) program is a handy
       way to save the output of e2fsck  to a file.

       It is also useful to send the output of dumpe2fs(8).  If a
       specific  inode  or  inodes  seems to be giving e2fsck
       trouble, try running the debugfs(8) command and send the
       output  of the stat(1u) command run on the relevant inode(s).
       If the inode is a directory, the debugfs  dump  command
       will allow you to extract the contents of the directory
       inode, which can sent to me after being first run  through
       uuencode(1).

       Always  include  the full version string which  e2fsck
       displays when it is run, so I know which version you are
       running.

AUTHOR
       This  version  of  e2fsck  was  written  by  Theodore  Ts’o
       <tytso@mit.edu>.

SEE ALSO
       mke2fs(8), tune2fs(8), dumpe2fs(8), debugfs(8)

E2fsprogs version 4         1.18     November 1999
```

C The GNU General Public License

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

< one line to give the program’s name and a brief idea of what it does.> Copyright (C) 19yy < name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’. This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

< signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

D Important keys

A short summary of helpful and important keys.

Enter	Executes a command on the command line; otherwise, a linefeed.
Alt + F1-F6	Change to another console (in text mode).
Ctrl + Alt + F1-F6	Under X11 and Dosemu: change to one of the text consoles.
Alt + F7	Switch from a text console back to X11.
Ctrl + Alt + ⟵	Exit X11.
Shift ⇑ + Page ↑	Shows previous screens on console. Works until consoles are switched.
Shift ⇑ + Page ↓	Opposite of Shift ⇑ + Page ↑.
↑	Displays the previous commands on the command line.
↓	Just the opposite of ↑ (in the shell).
Shift ⇑ + Alt + Cursor	X11: Change to the next virtual desktop (in the direction of the cursor). The number of virtual desktops is set in `.fvwm[2]rc` with the parameter "DeskTopSize".
Ctrl + Alt + ±(gray)	Changes the screen resolution according to entries in *XF86Config*.
Ctrl+D	Log out. Corresponds to **`exit`**. EOF (end of file). Affected by entry in `/etc/profile`: **`ignoreeof=x`**. Here, **`x`** indicates how many times command can be repeated until executed.

Table D.1:

Bibliography

[Alm94] Almesberger, Werner: *LILO User's guide*, 1994. – (see file `/usr/doc/lilo/user.dvi`)

[Bai97] Bailey, Edward C.: *Maximum RPM*. Red Hat, 1997. – (ISBN 1-888172-78-9)

[CAR93] Costales, Bryan; Allman, Eric ; Rickert, Neil: *sendmail*. O'Reilly & Associates, Inc., 1993. – (ISBN 1-56592-056-2)

[CB96] Cheswick, William R.; Bellovin, Steven M.: *Firewalls und Sicherheit im Internet*. Addison Wesley GmbH, 1996. – (ISBN 3-89319-875-x)

[CR91] Cameron, Debra; Rosenblatt, Bill: *Learning GNU Emacs*. O'Reilly & Associates, Inc., 1991. – (ISBN 0 937175-84-6)

[CZ96] Chapman; Zwicky: *Einrichten von Internet Firewalls. Sicherheit im Internet gewährleisten.*. O'Reilly & Associates, Inc., 1996. – (ISBN 3-930673312)

[Daw95] Dawson, Terry: *Linux NET-2/NET-3 HOWTO*, v2.8, 07 Jan 1995. – (see file `/usr/doc/howto/NET-2-HOWTO`)

[FCR93] Fang, Chin; Crosson, Bob ; Raymond, Eric S.: *The Hitchhiker's Guide to X386/XFree86 Video Timing (or, Tweaking your Monitor for Fun and Profit)*, 1993. – (see file `/usr/X11/lib/X11/doc/VideoModes.doc`)

[Fri93] Frisch, Æleen: *Essential System Administration*. O'Reilly & Associates, Inc., 1993. – (ISBN 0-937175-80-3)

[Gil92] Gilly, Daniel: *UNIX in a nutshell: System V Edition*. O'Reilly & Associates, Inc., 1992. – (ISBN 1-56592-001-5)

[GMS93] Goossens, Michel; Mittelbach, Frank ; Samarin, Alexander: *The LaTeX Companion*. Addison Wesley GmbH, 1993. – (ISBN 3-54199-8)

[Gri94] Grieger, W.: *Wer hat Angst vorm Emacs?*. Addison Wesley GmbH, 1994. – (ISBN 3-89319-620-X)

[GS93] Garfinkel, Simson; Spafford, Gene: *Practical UNIX Security*. O'Reilly & Associates, Inc., 1993. – (ISBN 0-937175-72-2)

[Her92] Herold, H.: *UNIX Grundlagen*. Addison Wesley GmbH, 1992. – (ISBN 3-89319-542-8)

[HHMK96]Hetze, Sebastian; Hohndel, Dirk; Müller, Martin ; Kirch, Olaf: *Linux Anwenderhandbuch*. 6. LunetIX Softfair, 1996. – (ISBN 3-929764-05-9)

[Hof97] Hoffmann, Erwin: EMail-Gateway mit qmail. In: *iX* 12 (1997), S. 108ff

[Hun95] HUNT, Craig: *TCP/IP Netzwerk Administration*. O’Reilly & Associates, Inc., 1995. – (ISBN 3-930673-02-9)

[Kie95] KIENLE, Micheal: TIS: Toolkit für anwendungsorientierte Firewall-Systeme. In: *iX* 8 (1995), S. 140ff

[Kir95] KIRCH, Olaf: *LINUX Network Administrator’s Guide*. O’Reilly & Associates, Inc., 1995. – (ISBN 1-56592-087-2)

[Kof95] KOFLER, M.: *Linux*. Addison Wesley GmbH, 1995. – (ISBN 3-89319-796-6)

[Kop94] KOPKA, Helmut: *LaTeX-Einführung*. Addison Wesley GmbH, 1994. – (ISBN 3-89319-664-1)

[Kun95] KUNITZ, Ulrich: Sicherheit fast kostenlos: Einrichtung eines kostenlosen Firewall-Systems. In: *iX* 9 (1995), S. 176ff

[Lam90] LAMB, Linda: *Learning the vi Editor*. O’Reilly & Associates, Inc., 1990. – (ISBN 0-937175-67-6)

[Lam94] LAMPORT, Leslie: *LaTeX User’s Guide and Reference Manual*. Addison Wesley GmbH, 1994. – (ISBN 0-201-52983-1)

[Lef96a] LEFFLER, Sam: *HylaFAX Home Page*, 1996

[Lef96b] LEFFLER, Sam: *TIFF Software*, 1996

[OT92] O’REILLY, Tim; TODINO, Grace: *Manging UUCP and Usenet*. O’Reilly & Associates, Inc., 1992. – (ISBN 0-937175-93-5)

[Per94] PERLMAN, G.: *Unix For Software Developers*. Prentice-Hall, 1994. – (ISBN 13-932997-8)

[Pug94] PUGH, K.: *UNIX For The MS-DOS User*. Prentice-Hall, 1994. – (ISBN 13-146077-3)

[SB92] SCHOONOVER, M.; BOWIE, J.: *GNU Emacs*. Addison Wesley GmbH, 1992. – (ISBN 0-201-56345-2)

[Sch98] SCHEIDERER, Jürgen: Sicherheit Kostenlos - Firewall mit Linux. In: *iX* 12 (1998)

[Sto98] STOLL, Clifford: *Kuckucksei; Die Jagd auf die deutschen Hacker, die das Pentagon knackten*. Fischer-TB.-Vlg., 1998. – (ISBN 3596139848)

[The96] THE XFREE86™-TEAM: *XF86Config(4/5) - Configuration File for Xfree86™*, 1996. – Manual-Page zu XFree86™

[TSP93] TODINO, Grace; STRANG, John ; PEEK, Jerry: *Learning the UNIX operating system*. O’Reilly & Associates, Inc., 1993. – (ISBN 1-56592-060-0)

[Wel94] WELSH, Matt: *Linux Installation and Getting Started*. 2. S.u.S.E. GmbH, 1994. – (ISBN 3-930419-03-3)

[WK95] WELSH, Matt; KAUFMAN, Lars: *Running Linux*. O’Reilly & Associates, Inc., 1995. – (ISBN 1-56592-100-3)