SUSE® LINUX® Administration®

COURSE 3037

Novell Training Services

www.novell.com

AUTHORIZED COURSEWARE

Volume 1

Part # 100-005040-001
Version 3

Novell®

Proprietary Statement

Novell, Inc.
1800 South Novell Place
Provo, UT 84606-2399

Disclaimer

Novell, Inc. makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc. reserves the right to revise this publication and to make changes in its content at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any NetWare software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc. reserves the right to make changes to any and all parts of NetWare software at any time, without obligation to notify any person or entity of such changes.

This Novell Training Manual is published solely to instruct students in the use of Novell networking software. Although third-party application software packages are used in Novell training courses, this is for demonstration purposes only and shall not constitute an endorsement of any of these software applications.

Further, Novell, Inc. does not represent itself as having any particular expertise in these application software packages and any use by students of the same shall be done at the students' own risk.

Software Piracy

Throughout the world, unauthorized duplication of software is subject to both criminal and civil penalties.

If you know of illegal copying of software, contact your local Software Antipiracy Hotline.

For the Hotline number for your area, access Novell's World Wide Web page at http://www.novell.com and look for the piracy page under "Programs."

Or, contact Novell's anti-piracy headquarters in the U.S. at 800-PIRATES (747-2837) or 801-861-7101.

Trademarks

Novell, Inc. has attempted to supply trademark information about company names, products, and services mentioned in this manual. The following list of trademarks was derived from various sources.

Novell, Inc. Trademarks

Novell, the Novell logo, NetWare, BorderManager, ConsoleOne, DirXML, GroupWise, iChain, ManageWise, NDPS, NDS, NetMail, Novell Directory Services, Novell iFolder, Novell SecretStore, Ximian, Ximian Evolution and ZENworks are registered trademarks; CDE, Certified Directory Engineer and CNE are registered service marks; eDirectory, Evolution, exteNd, exteNd Composer, exteNd Directory, exteNd Workbench, Mono, NIMS, NLM, NMAS, Novell Certificate Server, Novell Client, Novell Cluster Services, Novell Distributed Print Services, Novell Internet Messaging System, Novell Storage Services, Nsure, Nsure Resources, Nterprise, Nterprise Branch Office, Red Carpet and Red Carpet Enterprise are trademarks; and Certified Novell Administrator, CNA, Certified Novell Engineer, Certified Novell Instructor, CNI, Master CNE, Master CNI, MCNE, MCNI, Novell Education Academic Partner, NEAP, Ngage, Novell Online Training Provider, NOTP and Novell Technical Services are service marks of Novell, Inc. in the United States and other countries. SUSE is a registered trademark of SUSE Linux AG, a Novell company. For more information on Novell trademarks, please visit http://www.novell.com/company/legal/trademarks/tmlist.html.

Other Trademarks

Adaptec is a registered trademark of Adaptec, Inc. AMD is a trademark of Advanced Micro Devices. AppleShare and AppleTalk are registered trademarks of Apple Computer, Inc. ARCserv is a registered trademark of Cheyenne Software, Inc. Btrieve is a registered trademark of Pervasive Software, Inc. EtherTalk is a registered trademark of Apple Computer, Inc. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. Linux is a registered trademark of Linus Torvalds. LocalTalk is a registered trademark of Apple Computer, Inc. Lotus Notes is a registered trademark of Lotus Development Corporation. Macintosh is a registered trademark of Apple Computer, Inc. Netscape Communicator is a trademark of Netscape Communications Corporation. Netscape Navigator is a registered trademark of Netscape Communications Corporation. Pentium is a registered trademark of Intel Corporation. Solaris is a registered trademark of Sun Microsystems, Inc. The Norton AntiVirus is a trademark of Symantec Corporation. TokenTalk is a registered trademark of Apple Computer, Inc. Tru64 is a trademark of Digital Equipment Corp. UnitedLinux is a registered trademark of UnitedLinux. UNIX is a registered trademark of the Open Group. WebSphere is a trademark of International Business Machines Corporation. Windows and Windows NT are registered trademarks of Microsoft Corporation.

All other third-party trademarks are the property of their respective owners.

Contents

SECTION 3 Manage the Linux File System

SECTION 5 Manage System Initialization

SECTION 9 Enable Internet Services

Introduction

In the *SUSE Linux Administration* (3037) course, you learn basic SUSE Linux Enterprise Server 9 (SLES 9) administration skills.

These skills, along with those taught in the *SUSE Linux Fundamentals* (3036) and *SUSE Linux Advanced Administration* (3038) courses, prepare you to take the Novell® Certified Linux® Professional (Novell CLP) certification practicum test.

The contents of your student kit include the following:

- SUSE Linux Administration Manual - Volume 1
- SUSE Linux Administration Manual - Volume 2
- SUSE Linux Administration Course CD
- SLES 9 VMware Server DVD
- SUSE Linux Enterprise Server 9 CDs (CD 1 – CD 6)

The *SLES 9 VMware Server DVD* contains a VMware Workstation SLES 9 server that you can use with the *SUSE Linux Administration Self-Study Workbook* (in PDF format on your Course CD) outside the classroom to practice the skills you need to take the Novell CLP practicum.

Instructions for setting up a self-study environment are included in the *SUSE Linux Administration Self-Study Workbook*.

If you do not own a copy of VMware Workstation, you can obtain a 30-day evaluation version at www.vmware.com. If you want to dedicate a machine to install SLES 9, instructions are also provided in the Self-Study Workbook.

Course Objectives

This course teaches you how to perform the following Linux system administration tasks for SLES 9:

- Update and check the health of a SLES 9 server
- Perform administrative tasks with YaST
- Manage users and groups
- Provide basic system security
- Manage the Linux file system
- Manage software installation
- Manage system initialization, processes, and services
- Connect the server to the network
- Provide basic network services (such as printing and web access)
- Remotely access a SLES 9 server

These are administrative skills common to an entry-level administrator or help desk technician in an enterprise environment.

The last half of the final day of class is reserved for building a SLES 9 solution on your own to prepare for taking the Novell CLP Practicum.

Audience

While the primary audience for this course is the current Novell CNE℠, Linux professionals, and administrators with experience in other operating systems can also use this course to help prepare for the Novell CLP Practicum.

Certification and Prerequisites

This course helps you prepare for the Novell Certified Linux Professional (Novell CLP) Practical Test, called a *practicum*. The Novell CLP is an entry-level certification for people interested in becoming Linux administrators.

As with all Novell certifications, course work is never required. You only need only pass a Novell CLP Practicum (050-689) in order to achieve the certification.

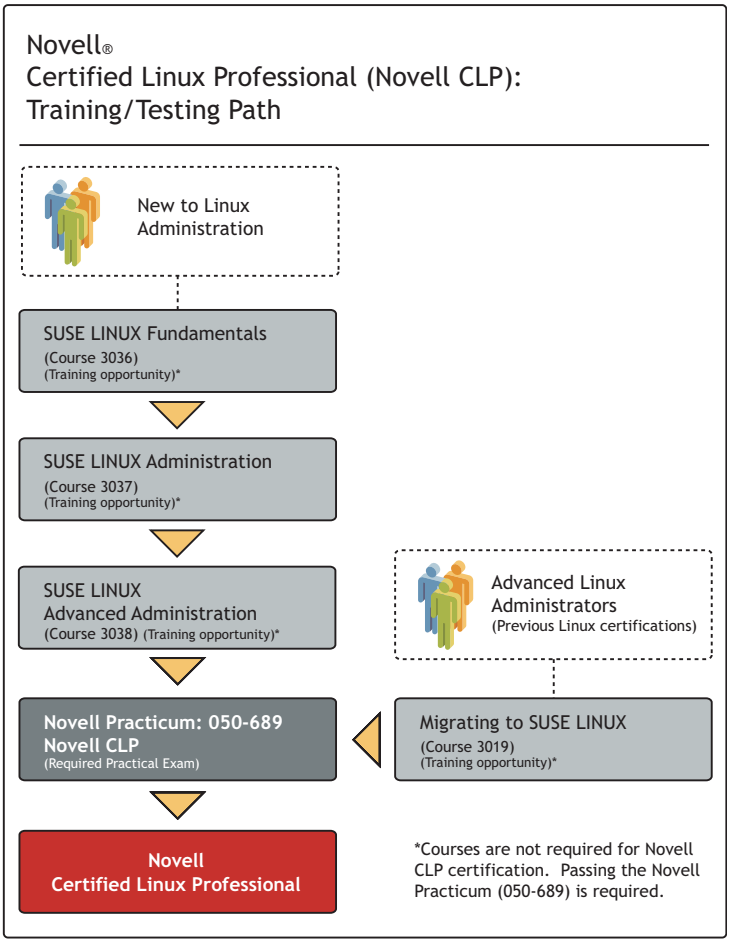
The Novell CLP Practicum is a hands-on, scenario-based exam where you apply the knowledge you have learned to solve real-life problems—demonstrating that you know what to do and how to do it.

The practicum tests you on objectives in this course (Linux Administration - Course 3037) and the skills outlined in the following Novell CLP courses:

- Linux Fundamentals - Course 3036
- Advanced Linux Administration - Course 3038

The following illustrates the training/testing path for Novell CLP:

Figure Intro-1



Before attending this course, you should have completed Linux Fundamentals — 3036, or have experience in the technical skills and knowledge identified by the learning objectives of that course.

For more information about Novell certification programs and taking the Novell CLP Practicum, see http://www.novell.com/education/certinfo.

SLES 9 Support and Maintenance

The copy of SUSE Linux Enterprise Server 9 (SLES 9) you receive in your student kit is a fully functioning copy of the SLES 9 product.

However, to receive official support and maintenance updates, you need to do one of the following:

- Register for a free registration/serial code that provides you with 30 days of support and maintenance.
- Purchase a copy of SLES 9 from Novell (or an authorized dealer).

You can obtain your free 30-day support and maintenance code at http://www.novell.com/products/linuxenterpriseserver/eval.html.

You will need to have or create a Novell login account to access the 30-day evaluation.

SLES 9 Online Resources

Novell provides a variety of online resources to help you configure and implement SLES 9.

These include the following:

- http://www.novell.com/products/linuxenterpriseserver/

  This is the Novell home page for SLES 9.

- http://www.novell.com/documentation/sles9/index.html

  This is the Novell Documentation web site for SLES 9.

- http://support.novell.com/linux/

  This is the home page for all Novell Linux support, and includes links to support options such as the Knowledgebase, downloads, and FAQs.

- http://www.novell.com/coolsolutions

  This Novell web site provides the latest implementation guidelines and suggestions from Novell on a variety of products, including SUSE Linux.

Agenda

The following is the agenda for this 5-day course:

Table Intro-1

	Section	Duration
Day 1	**Introduction**	00:30
	Section 1: Introduction to Managing the SUSE Linux Enterprise Server	02:00
	Section 2: Manage User Access and Security	03:00

(cont.) **Table Intro-1**

	Section	Duration
Day 2	**Section 3:** Manage the Linux File System	03:00
	Section 4: Manage Software for SUSE Linux Enterprise Server	02:00
	Section 5: Manage System Initialization	02:00
Day 3	**Section 6:** Manage Linux Processes and Services	03:00
	Section 7: Connect the SUSE Linux Enterprise Server to the Network	03:00
Day 4	**Section 8:** Enable Infrastructure Services	03:00
	Section 9: Enable Internet Services	03:00
Day 5	**Section 10:** Manage Remote Access	02:00
	Section 11: LiveFire Exercise	04:00

Scenario

As system administrator for your Digital Airlines office, you have been tasked by the company to migrate several network services to SLES 9 servers over the next year.

As part of the rollout plan, you would like to install SLES 9 on a prototype/staging server that you can use to do the following:

- Become familiar with basic administrative tasks on the local host (such as providing user access and security)
- Connect to the network to test a variety of services you will be migrating (such as file and print)
- Provide limited access for training others in your office (such as the database group) who will be using or configuring these services

- Test updating and remote administration of SLES 9

Once you complete this initial testing of services and administrative tasks, you will then be in a position to begin rolling out SLES 9 according to guidelines from Digital Airlines corporate headquarters.

Exercise Conventions

When working through an exercise, you will see conventions that indicate information you need to enter that is specific to your server.

The following describes the most common conventions:

- ***italicized/bolded text***. This is a reference to your unique situation, such as the host name of your server.

  For example, if the host name of your server is DA50, and you see the following,

  ***hostname*.digitalairlines.com**

  you would enter

  DA50.digitalairlines.com

- **10.0.0.*xx***. This is the IP address that is assigned to your SLES 9 server.

  For example, if your IP address is 10.0.0.50, and you see the following

  10.0.0.*xx*

  you would enter

  10.0.0.50

- **Select.** The word *select* is used in exercise steps to indicate a variety of actions including clicking a button on the interface and selecting a menu item.

- **Enter and Type.** The words *enter* and *type* have distinct meanings.

  The word *enter* means to type text in a field or at a command line and press the Enter key when necessary. The word *type* means to type text without pressing the Enter key.

  If you are directed to type a value, make sure you do not press the Enter key or you might activate a process that you are not ready to start.

SECTION 1 Introduction to Managing the SUSE Linux Enterprise Server

In this section you learn about the features of SUSE Linux Enterprise Server 9 (SLES 9), how to use YaST, and how to monitor your current installation.

Objectives

1. Describe SLES 9
2. Access and Use YaST
3. Monitor Your SLES 9 System

Objective 1 Describe SLES 9

Backed by an extensive Novell support infrastructure and partner network, SLES 9 is a secure, reliable platform for open source computing in the enterprise.

The new 2.6 kernel, scalability and availability, management tools and developer tools make SLES 9 the most flexible, most convenient, most scalable version yet.

SLES 9 also provides open application programming interfaces (APIs) and other development tools that simplify Linux* integration and customization.

And, since SLES 9 is backed by Novell, you can deploy it with confidence, knowing that you've got the help of hundreds of software engineers, support staff, and consultants who are dedicated to Linux, open source, and customer success.

In this objective you learn the following about SLES 9:

- What's New in SLES 9
- Hardware Support and System Requirements
- Service and Support
- Linux Standards and SLES 9

For a complete list of SLES 9 technical features, download the file *sles9_featurelist_technical.pdf* at http://www.suse.de/en/business/products/server/sles/index.html.

What's New in SLES 9

Although the list of exclusive and open source features in SLES 9 is extensive, knowing what's new in SLES 9 helps you understand the possibilities for integrating or migrating to a SUSE Linux Enterprise Server environment.

The following are feature categories that cover the new functionality of SLES 9:

- New Kernel
- New Scalability and Available Features
- New Systems Management Tools
- New Developer Tools
- New Software

Many of these new features are advanced topics and are not discussed in this basic Linux Administration course.

However, by becoming familiar with these technical features, you are in a better position to meet requirements and plans for introducing SLES 9 into your enterprise network environment.

New Kernel

SLES 9 includes the Linux kernel version 2.6, a dramatic improvement over earlier systems in terms of scale, speed, and power.

USB 2.0 and Bluetooth are supported, Hotplug support exists for SCSI, USB, Firewire, PCI, and CPU (the latter only on PowerPC). The support of Ipv6 has improved, and the ALSA sound system is now part of the kernel.

The following is an overview of improvements directly related to the new kernel:

Table 1-1

Feature	Description
Kernel 2.6.*x* scalability enhancements	These enhancements include the following: ■ **More processors:** More than 128 CPUs have been tested on available hardware, but theoretically, there is no limit on the number that will work. ■ **More users:** Systems can now have more than 4 billion unique users. ■ **More processes:** Run up to 65,535 user-level processes, plus additional kernel-level processes that represent threads. ■ **More open files:** SLES 9 automatically tunes resource usage dynamically to support the maximum number of simultaneous open files.
Kernel 2.6 device enhancements	These enhancements include the following: ■ **More device types:** 4,095 major device types and more than a million subdevices per type make larger storage arrays, print farms, and tape units more feasible. ■ **More devices:** The server can now manage more devices. For example, it can control up to 32,000 SCSI disks. ■ **Faster devices:** Support for high-speed USB 2.0 and Firewire* (IEEE 1394 and 1394b). ■ **Higher throughput:** High-speed Serial ATA (S-ATA) device support enables transfer rates up to 150 MB/sec.

(continued) **Table 1-1**

Feature	Description
Non-Uniform Memory Access (NUMA)	NUMA lets SUSE Linux Enterprise Server scale more efficiently for systems with dozens or hundreds of CPUs because CPUs can access a dedicated memory bus for local memory. It also supports multiple interconnected memory nodes, each supporting a smaller number of CPUs. The result is greater scalability for applications that use local memory.
NUMA development tools	For x86-64 (Opteron) and IA-64 (Itanium*), the NUMA tools allow developers to fine-tune applications for NUMA usage. Both Oracle and DB2 are developing NUMA API support, and Oracle already uses it in testing—one reason that SUSE Linux Enterprise Server has surpassed other distributions in database benchmarks.
Hyperthreading	Hyperthreading enables multithreaded server software applications to execute threads in parallel within each individual server processor, dramatically improving transaction rates and response times.

(continued) **Table 1-1**

Feature	Description
Flexible I/O scheduler	The new I/O scheduler lets you tune the server to match its usage with four I/O behavior policies: ■ **Complete Fair Queuing:** CFQ is suitable for a wide variety of applications, especially desktop and multimedia workloads. It is the default I/O scheduler. CFQ treats all competing processes equally by assigning each process a unique request queue and giving each queue equal bandwidth. ■ **Deadline:** The deadline I/O scheduler implements a per-request service deadline to ensure that no requests are neglected. Deadline policy is best for disk-intensive database applications. ■ **Anticipatory:** The anticipatory I/O scheduler uses the deadline mechanism plus a heuristic to anticipate the actions of applications. This provides greater disk throughput but slightly increases latency. The anticipation heuristic is suitable for file servers but does not work as well for database workloads. ■ **No-Op:** This “no-operation” mode does no sorting and is used only for disks that perform their own scheduling or are randomly accessible. The first three behaviors group and merge requests to maximize request sizes, cutting down on the amount of seeking performed.
Class-based Kernel Resource Management (CKRM)	CKRM lets you provide differentiated service at a user or job level and prevent denial-of-service attacks. It also increases the accuracy of resource-consumption metering.

New Scalability and Available Features

Delivering rapid scalability and high availability, SLES 9 is well suited to meet the demands of the datacenter.

SLES 9 provides the rapid scalability and high performance systems that today's businesses require, and includes the following:

Table 1-2

Feature	Description
Heartbeat high-availability system	The Heartbeat system provides core cluster membership and messaging infrastructure. It implements the Open Clustering Framework APIs (a subset of the Service-Availability Forum APIs) to provide low-level services for node fencing, fault isolation, and basic two-node failover. The node failure detection time can be tuned to under one second, allowing for subsecond failovers in some environments. In the case of a node failure, the Heartbeat system checks I/O to ensure data integrity, then moves resources to the alternate node. Return to action of failed nodes can be set to automatic or manual, depending on your preference.
Enterprise Volume Management System (EVMS)	EVMS lets you handle storage through one mechanism. RAID, LVM, various file system formats, disk checking and maintenance, bad block relocation and more are handled by a single tool.
Distributed Replicated Block Device (DRBD)	This networked disk-management tool constructs single partitions from multiple disks that mirror each other. It is similar to a RAID1 system but runs over a network. The partition size can be changed at runtime.

New Systems Management Tools

SLES 9 simplifies system administration. The following are the new systems management tools available:

Table 1-3

Feature	Description
Novell ZENworks® Linux Management support	Novell ZENworks Linux Management lets you deploy software enterprise-wide. It includes Red Carpet Daemon in the box, and specifically distributes YaST patches and updates from Novell and SUSE Linux.
Novell ZENworks Linux Management server support	Novell ZENworks Linux Management includes the bundled SUSE Linux Enterprise Server and Novell ZENworks Linux Management. It enables onsite hosting of the server and offers granular administration, with tight control over software and patch deployment. It also provides integration into IT Service Management (such as ITIL) change-control processes and offers scriptable or Web-based administration.
New YaST Modules	YaST has several new modules in SLES 9, including the following: ■ A mail server configuration tool that lets you create secure servers with IMAP and POP service, quotas, access control lists, name spaces, routing, local mail delivery, server-side filtering of viruses and junk mail, and other enterprise-level mail system features. ■ A VPN configuration assistant for both client and server. The VPN is compatible with Linux and Windows clients and can be configured without additional software. ■ Full Samba 3 configuration.

(continued) **Table 1-3**

Feature	Description
Improved YaST Tools	The following YaST configuration tools have been enhanced and updated in this release, and range from user-interface improvements to new features and capabilities: ■ Improved network-configuration tools, including DNS, DHCP, LDAP, NIS, Postfix, and TFTP ■ NFS and Samba network file system settings ■ Automated default certificate authority (CA) for servers, including LDAP, Apache*, and Postfix ■ Virtual Private Network (VPN) ■ Installation server ■ Boot server ■ CD creation ■ User-Mode Linux installation and virtualization setup ■ Apache ■ Wake on LAN ■ High-availability tools expanded to work with Heartbeat ■ Update server ■ User-management-tool support for plugins for external back-ends, including IMAP and Samba

New Developer Tools

Developers trust SUSE Linux Enterprise Server to supply a dependable platform and offer rich software-development capabilities through built-in network services and protocols, including CUPS, DNS, DHCP, IMAP, NTP, SLP, Postfix, PXE, Proxy, Samba, SNMP and SMTP.

SLES 9 also includes application and database services and supports popular solutions from hundreds of vendors.

The following are new developer tools included in SLES 9:

Table 1-4

Feature	Description
C# and .NET	The SUSE Linux SDK has always provided state-of-the-art Linux software-development tools for a stable, multiplatform codebase. The newest addition to the toolbox, the Mono™ project, lets developers build and run .NET applications on Linux and other operating systems. Application frameworks supported include ASP.NET, ADO.NET and gtk#.

New Software

Like its predecessors, SLES 9 supports the best of new software as well as updated favorites. These include the following:

Table 1-5

Feature	Description
Samba 3	The latest version of the Samba package supports Unicode* and Active Directory* and offers improved authentication and printing tools. Other improvements include migration tools, support for establishing trust relationships with Windows NT* 4.0 domain controllers, improved ACL settings, and better performance.

(continued) **Table 1-5**

Feature	Description
User-Mode Linux	User-Mode Linux (UML), originally developed as a kernel debugging tool, is a tool that allows a Linux instance to run as a regular process under Linux. In other words, UML makes it possible to run several instances of Linux at once. UML is the virtualization tool of choice for the Linux kernel and VPN development and can be used to provide extra systems in labs, testing environments or hosting providers. It also makes an excellent secondary firewall. After installing the UML package like any other application binary, you can create additional Linux instances that are installed into directories and managed using the same tools and applications as the non-virtual instance of Linux. While virtual performance is not as optimal as its non-virtual counterpart, and although UML systems require significant amounts of memory, UML instances can generally be used for the same tasks as nonvirtual Linux instances.

Hardware Support and System Requirements

The following are hardware support and system requirement specifications for SLES 9:

- Supported Processor Platforms
- Supported Hardware Extensions
- SLES 9 Hardware Requirements

To avoid unnecessary waste of resources, see http://cdb.suse.de/ to find out if a particular piece of hardware is supported by SLES 9 before you buy it or try to get it to work.

Supported Processor Platforms

The following are supported processor platforms:

Table 1-6

Feature	Description
*x*86	The *x*86 platform, the most common personal computer hardware platform, is the basis for chips from Intel, AMD, VIA and Transmeta. Supported processors include Intel Pentium and Xeon processors, AMD Athlon and K7 series, and the AMD Opteron and Athlon64 chipsin 32-bit mode.
*x*86-64 (AMD-64)	The *x*86-64 architecture was created by AMD and is a 64-bit extension of the x86 platform that also runs 32-bit legacy code. Supported processors include the AMD Athlon64 and Opteron, and the forthcoming Intel Xeon EM64T. Note that not all 32-bit applications are certified to run perfectly in this environment; check with your ISV or perform extensive testing before deployment.

(continued) **Table 1-6**

Feature	Description
Itanium Processor Family	SUSE Linux Enterprise Server supports the Itanium processor family, a 64-bit platform from Intel and HP that includes an emulator for x86 32-bit hardware. Because 32-bit support relies on an emulator, 32-bit performance can suffer. However, 64-bit performance benefits from a clean break with the *x*86 platform.
IBM POWER (iSeries and pSeries systems)	SLES 9 is available for 64-bit IBM POWER series hardware. The IBM POWER 64-bit architecture is used in systems from Apple and IBM. It supports execution of code built for the 32-bit edition of the POWER platform. POWER systems include IBM iSeries and pSeries servers.
IBM zSeries (S/390*x*)	The IBM zSeries is a 64-bit platform mostly used in the S/390*x* mainframe series. zSeries hardware can also run code built for the earlier 31-bit S/390 systems. SUSE Linux Enterprise Server running on zSeries hardware can be used for zVM and LPAR virtualization of both 31-bit and 64-bit systems.

(continued) **Table 1-6**

Feature	Description
IBM S/390 (31-bit)	The IBM S/390 architecture is an anomaly in that it uses 31 bits, rather than the more common thirty-two. In these systems, the thirty-second processor bit is used by the chip itself instead of being made available to the software. SLES 9 is available in a 31-bit compile for S/390 mainframes. SLES 9 on S/390 hardware can be used for zVM and LPAR virtualization of 31-bit systems only.

Supported Hardware Extensions

The following are supported hardware extensions for SLES 9:

Table 1-7

Feature	Description
InfiniBand (Exclusive)	InfiniBand technology is used for intersystem and interprocess communications (IPC) within a single system. IPC is used in parallel clustering systems, where it provides greater performance, lower latency, faster data sharing, improved usability, and built-in security and reliability.
USB 2.0	USB 2.0 is a powered connection with bandwidth of up to 480 megabits per second.
Firewire (IEEE 1394)	Firewire (IEEE 1394) is a powered connection with bandwidth of up to 400 megabits per second.

(continued) **Table 1-7**

Feature	Description
ACPI	The ACPI system handles low-level hardware control and configuration, including power management, processor speed and temperature management.

SLES 9 Hardware Requirements

The following are hardware requirements for SLES 9:

Table 1-8

Feature	Description
For Installation	The following are memory requirements for installation: ■ **Local Installation:** 256 MB RAM ■ **SSH-based network install, graphical:** 256 MB RAM ■ **VNC-based network install using FTP:** 512 MB RAM
For Operation	The following are minimum requirements for running the SLES 9 operating system: ■ 256 MB RAM ■ 500 MB hard-disk space for software
Recommended	The following are general recommendations for running services on SLES 9: ■ A Pentium® III or AMD 750 Mhz or faster computer ■ 512 MB to 3 GB RAM, at least 256 MB per CPU ■ 4 GB hard-disk space ■ Network interface (Ethernet, wireless, or modem)

(continued) **Table 1-8**

Feature	Description
Suggestions for specific uses	The following are recommendations for specific services: ■ **Print servers:** If rendering is done on a server, a faster processor or additional processors. ■ **Web servers:** Additional RAM can improve caching. Additional processors will improve web application performance. ■ **Database server:** Additional RAM can improve caching. Using multiple disks permits parallel I/O. ■ **File servers:** Additional disks or a RAID system can improve I/O throughput.

Service and Support

Many of you have relied on Novell to provide the best enterprise level services in the industry. Now you can depend on Novell to provide that same level of service for Linux technologies provided by Novell.

- Support Programs
- Bug Fixes and Security Patches
- Maintenance Contract

Support Programs

The following are current support programs provided by Novell:

- **Premium Service.** Premium Service provides the high level of customized service needed for mission critical systems and covers all your Novell products and technologies.

You choose the level of support that makes the most sense for your business—from occasional telephone support to dedicated support engineers who bring full time support, knowledge, and expertise to your organization for business critical or highly customized solutions.

- **SUSE Linux Server Support.** SUSE Linux Server Support lets you access Novell's expertise for the ongoing support of a single SUSE Linux server.
- **Novell Linux Small Business Support.** The Novell Linux Small Business Support program supports a range of Novell Linux products, offering direct access to Novell Linux support experts, fast response times, and additional support resources for preventing and resolving technical issues.
- **Remote and Managed Services.** Reduce system down time, control your costs, and free up your IT resources with Novell's Remote and Managed Services.

  Novell's experts can proactively monitor and manage all your Novell technologies, freeing up your IT staff for more strategic and profitable projects.
- **Novell Technical Subscriptions.** These subscriptions make it easier to manage today's complex networks. By subscribing, you'll receive the latest information and resources, including new Novell software, advanced technical resources, exclusive online tools, and much more.
- **Online support options.** These include the Knowledgebase (TIDs), support forums, downloadable files, patches and drivers, product tips and tricks, documentation, and much more.

Additional free and fee-based services are available for issues not covered during the warranty period as well as for ongoing support of Novell's Linux consumer products.

You can access all of Novell's Linux support options at http://support.novell.com/linux/.

For a list of frequently asked questions (FAQs) about SLES 9 support, see http://support.novell.com/linux/linux_faq.html.

Bug Fixes and Security Patches

From a business perspective, it is important that an operating system and software is maintained over a long period of time. This means that security patches or bug fixes need to be available for several years.

Bug fixes and security patches for SLES 9 are provided for 5 years to those customers with a valid maintenance contract or registration code for updates.

This is not the case SUSE Linux products such as SUSE Linux Professional or Personal (the Linux distributions for the consumer market).

Maintenance Contract

Another aspect of product support is certification of the operating system by other software vendors. For example, Oracle and SAP do not support their software if it is run on an operating system not certified by them.

Such certifications are not done for the SUSE consumer products, but only for the SUSE Linux Enterprise Server.

Because maintenance and certifications require considerable know-how and manpower, this service is not provided for free.

This is the reason why there is a fee for a maintenance contract, despite the fact that the Linux kernel and most of the software that accompanies SLES 9 (such as Postfix and Apache) are contributed voluntarily by thousands of developers around the globe.

The maintenance contract assures an up-to-date product throughout its product life cycle.

Linux Standards and SLES 9

SLES 9 adheres to the following Linux standards:

- Linux Standard Base (LSB)
- File System Hierarchy Standard (FHS)
- TeX Directory Structure (TDS)

Linux Standard Base (LSB)

SUSE actively supports the efforts of the Linux Standard Base (LSB) project. The currently valid LSB specification is version 1.3.*x* and only covers the *x*86 architecture.

Apart from the File System Hierarchy Standard (FHS), which is now part of the LSB, the specification defines items such as the package format and details of the system initialization.

For up-to-date information about the LSB project, see http://www.linuxbase.org.

File System Hierarchy Standard (FHS)

In accordance with the LSB specification, SUSE Linux Enterprise Server is also compliant with the File System Hierarchy Standard or FHS (package *fhs*).

For this reason, in some cases it has been necessary in SUSE Linux Enterprise Server to move files or directories to their correct places in the file system, as specified by the FHS.

For example, one aim of the FHS is to define a structure in which */usr/* can be mounted as read-only.

For up-to-date information about FHS, see http://www.pathname.com/fhs/.

TeX Directory Structure (TDS)

TeX is a comprehensive typesetting system that runs on various platforms. It can be expanded with macro packages, like LaTeX, and consists of numerous files that must be organized according to the TeX Directory Structure (TDS).

teTeX is a compilation of current TeX software. On a SUSE Linux system, teTeX is installed in a way that ensures compliance with the requirements of both the TDS and the FHS.

For additional information on TDS, see ftp://ftp.dante.de/tex-archive/tds/.

Exercise 1-1 Explore Your KDE Desktop

When you first install SLES 9 and log in to a desktop environment, you probably want to take a few minutes to explore your desktop and check available resources.

In this course, SLES 9 has been installed for you, and you use the KDE desktop.

In this exercise, you log in as geeko (the normal user), and then explore and prepare your KDE desktop for performing the rest of the exercises in the course

In this (and other exercises), you switch between using Kate and vi as text editors. If you would prefer using vi, feel free to do so.

Do the following:

1. Log in to your SLES 9 server as **geeko** with a password of **N0v3ll** (a zero; not an uppercase O).

   Several messages appear during the initial login, including the following:

   - New Hardware found message
   - Welcome to SLES 9 message
   - "Warning" message about the powersave daemon not running or that you are not a member of the powersave system group
   - Kandolf's Useful Tips - KTip

2. From the New Hardware found message, deselect **Keep me informed about new hardware**.

3. Skip configuring the hardware device (such as a sound card) by selecting **No**.

4. Close the Welcome to SLES 9 message.

You can display this message at any time by selecting the **SUSE** icon on the KDE desktop.

5. From the Kandolf's Useful Tips dialog view some of the tips by selecting **Previous** or **Next**.

6. When you finish select **Show tips on startup** to deselect the option; then select **Close**.

   The Warning message about powersave appears for a few seconds each time you log in to the KDE desktop (by now it should have disappeared).

7. Disable the powersave Warning message by doing the following:

   a. Open a terminal window by selecting the **Terminal Program** icon on the panel at the bottom of the screen.

      A Tip of the Day - Konsole dialog appears.

   b. Select **Show tips on startup** to deselect the option; then select **Close**.

   c. From the terminal window, su (switch user) to root by entering **su -**; then enter a password of **novell**.

   d. Edit the file /opt/kde3/share/autostart/kpowersave.desktop by entering the following;

      vim /opt/kde3/share/autostart/kpowersave.desktop

      The file kpwersave.desktop appears in the vi editor.

   e. Scroll down to the bottom of the file (use the down arrow) until you find the following line:

      ```
      X-KDE-autostart-condition=kpowersaverc:General:Aut
      oStart:true
      ```

      You need to change the "true" value to "false."

   f. Press the Insert key; then make the change.

   g. When you finish, press **Esc**; then save the change and exit the vi editor by typing **:wq** and pressing **Enter**.

      You are returned to the command line.

8. View the vi tutorial by entering **vimtutor**.

A text file with several short lessons is opened in the vi editor. If you are new to vi or need to refresh your basic skills, try some of the lessons after finishing this exercise.

9. Close the vi editor without saving any changes by pressing **Esc**; then type **:q!** and press **Enter**.

10. (Optional) If you are connected to the Internet, you can test the connectivity from the command line by entering the following:

    ping -c 3 www.novell.com

    If there is no Internet access, you receive an "unknown host" message.

11. From the terminal window, check the IP address configured for eth0 by entering **ifconfig**.

    Because your network card is currently configured for automatic IP address setup through DHCP, you do not have an IP address assigned unless you are connected to a network that provides a DHCP server.

    In this case, the only IP address available is 127.0.0.1 (the localhost loopback address).

12. Check the SLES 9 help resources available by doing the following:

    a. From the panel at the bottom of the desktop, select the **SUSE HelpCenter** icon (the lifesaver).

       The SUSE HelpCenter provides a central location for viewing and searching many of the manuals installed with SLES 9.

    b. On the left, make sure the **Content** tab is selected, then select **SLES 9 Adminguide**.

       Notice the variety of topics available, including documentation for YaST administration tool (which you will be using in this course).

    c. Close the HelpCenter window.

d. From the desktop, select the **SUSE** icon.

   The Welcome to SUSE Linux Enterprise Server 9 dialog appears.

e. (Conditional) If you have Internet access, try selecting the **SUSE Hardware Database** link.

   The SUSE Hardware Database is especially critical for finding out if the hardware you have or the hardware you plan on purchasing is supported by SLES 9.

   There are also links on the left for finding out more about support issues.

f. (Conditional) If you have Internet access, try selecting the **Novell and Linux** link.

   This is Novell's home page for Linux products.

g. From the menu on the left, select **get involved > download**.

   This is Novell's download page for accessing the latest downloads for all products.

h. From the top of the page, select **SUPPORT > Knowledgebase**.

   From here you can access Novell's support database for information about SLES 9.

i. When you finish exploring, close all open windows.

13. Check you current hardware configuration against the hardware requirements for SLES 9:

14. requirements for SLES 9.

The following are general recommendations for running services on SLES 9

- A Pentium® III or AMD 750 Mhz or faster computer
- 512 MB to 3 GB RAM, at least 256 MB per CPU
- 4 GB hard-disk space
- Network interface (Ethernet, wireless, or modem)

Do the following:

a. From the panel at the bottom of the screen, select the **KDE Start Menu** icon (the green circle with the red N); then select **System > Monitor > Info Center**.

   The KDE Info Center appears.

   Use this tool to check your hardware configuration against the Recommended category in the SLES 9 Hardware Requirements table.

b. On the left, select categories such as **Processor**, **Memory**, **Storage Devices**, and **Network Interfaces**.

c. When you finish, close the KDE Info Center window.

15. Log out as geeko by selecting **KDE Menu > Logout > Logout**.

You are returned to the GUI login screen.

16. Log in again as **geeko** (password of **N0v3ll**) to test the changes you made to the desktop.

Notice that the messages no longer appear. If you left the terminal window open when logging out, a new terminal window opens for you when you log back in again.

The same Welcome message and other dialogs appear the first time you log in as any local user from the GUI login screen, and you will need to perform the same tasks to suppress the messages from appearing again.

(End of Exercise)

Objective 2 Access and Use YaST

YaST stands for Yet another Setup Tool. You can use YaST to complete many configuration tasks as a SUSE Linux Enterprise Server administrator.

To effectively use YaST for configuring SLES 9, you need to know the following:

- YaST Basics
- The Role of SuSEconfig

YaST Basics

Although you use YaST to install SLES 9, it has a much greater role as a system management and configuration tool.

You can use YaST to configure various services, to install software, to configure hardware, to manage users, and complete many other administrative tasks.

To perform basic administrative tasks with YaST, you need to know how to do the following:

- How to Start YaST With a GUI Interface
- How to Start YaST With a Text Interface (ncurses)

How to Start YaST With a GUI Interface

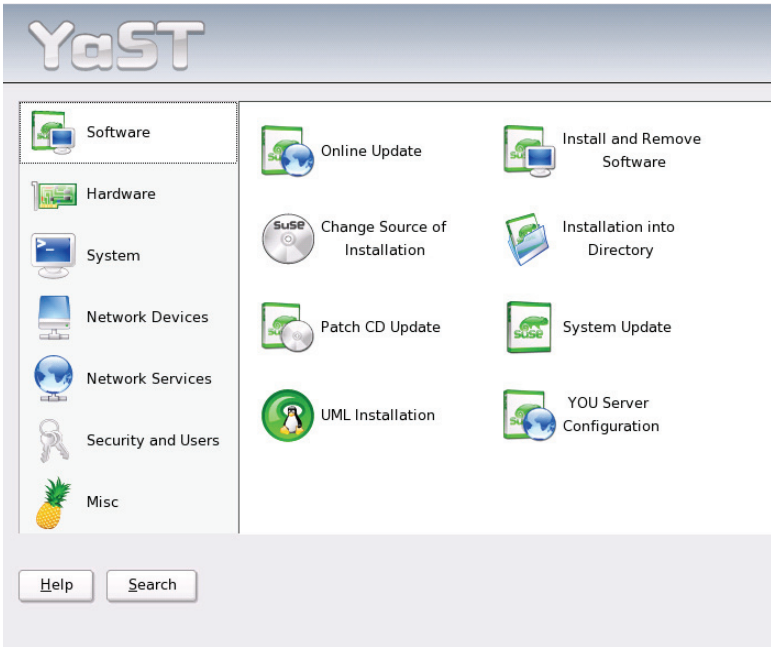
You can start YaST by doing one of the following:

- Select the **YaST** icon from the desktop, and then enter the root ***password*** (if you are not already logged in as root).
- From a terminal window, switch to root by entering **sux -** and the root ***password***; then enter **yast2**.

- Press **Alt+F2**, type **yast2**, select **Options**, select **Run as a different user**, and enter the root ***password***; then select **Run** (KDE only).

After using one of these three methods to start YaST, the following appears:

Figure 1-1



This is the main dialog of YaST, sometimes called the *YaST Control Center*.

From here you can select a category on the left (such as Software or System) and a module on the right (such as Online Update) to configure and manage your system from a GUI interface.

When you finish making changes with a YaST module, YaST uses backend services such as SuSEconfig (see "The Role of SuSEconfig" on 1-29) to implement the changes in the system.

Beside starting a module from the YaST Control Center, you can also start the module from a command line.

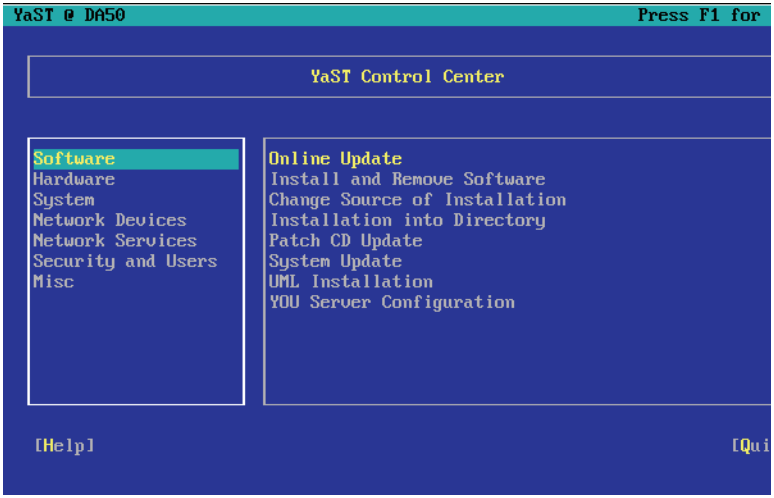
Make sure you are logged in to a terminal window as root with the command **sux -**; then list the names of the available modules by entering **yast2 -l**. Start a module by entering **yast2** ***module_name***.

How to Start YaST With a Text Interface (ncurses)

As root you can run YaST without a graphical user interface by entering **yast**. This starts the *ncurses* interface. If there is no X Windows running, the command yast2 also launches the ncurses interface.

The following is the YaST Control Center ncurses interface:

Figure 1-2



You can use the Tab key, Up and Down Arrow keys, or Alt+*letter* to navigate. Except for the navigation there is no functional difference between the YaST GUI and ncurses interface.

The Role of SuSEconfig

You can consider YaST as a front end to various other programs, such as a front end to RPM (RPM Package Manager) software management, a front end to user management, or a front end to various configuration files of different services (like a mail or web server).

Sometimes YaST writes the configuration changes you make directly into the final configuration file.

In many other cases there is an additional intermediate step, where the information you enter is first written to a file in the directory /etc/sysconfig/ and then written to its final destination.

This is where the program SuSEconfig becomes important.

SuSEconfig is a tool used in SUSE Linux Enterprise Server to configure the system according to the variables that are set in the various files in /etc/sysconfig/ and its subdirectories.

These files contain variables such as SYSLOGD_PARAMS="" in /etc/sysconfig/syslog and SMTPD_LISTEN_REMOTE="no" in */etc/sysconfig/mail*.

Some of these variables are used directly (such as in some start scripts). For example, if SYSLOGD_PARAMS is set to "-r," the daemon that logs system messages is directed to listen on port 514 for system messages from other hosts.

Other variables are used to modify other files. For example, if SMTPD_LISTEN_REMOTE is set to "yes," the variable INET_INTERFACES in /etc/postfix/main.cf is set to "all" by the script /sbin/SuSEconfig and the scripts in /sbin/conf.d/.

SuSEconfig acts as a back end for YaST and activates the configuration changes you make when using a YaST module.

If you modify files in /etc/sysconfig/ using an editor, all you might need to do is restart a service for the change to take place. However, you might also need to run SuSEconfig.

For this reason, we recommend running SuSEconfig after manually editing files in /etc/sysconfig/.

Additional details about SuSEconfig are provided throughout this and other SUSE Linux administration courses. You can also display the manual pages for SuSEconfig by entering **man suseconfig**.

Exercise 1-2 Customize Your SLES 9 Installation With YaST

In this exercise, you customize your SLES 9 installation using the YaST text user interface and the YaST graphical user interface.

Do the following:

1. Set the language and time zone with the YaST text user interface:

   During classroom setup, your SLES 9 server was configured to use US English.

   To check and change this setting with YaST (ncurses), do the following:

   a. From the KDE desktop, open a terminal window by selecting the **Terminal Program** icon from the bottom panel.

   b. Switch to the root user by entering **su -**; then enter a password of **novell**.

   c. Start YaST by entering **yast2**.

      The text version of the YaST Control Center appears with a list of categories on the left and a list of modules on the right.

   d. (Optional) You might want to expand the terminal window for a larger view of the YaST Control Center.

   e. From the YaST Control Center, select the **System** category and the **Choose Language** module.

      A Language selection dialog appears.

   f. Select ***your language***; then select **Accept** by pressing **Alt+A**.

      You are returned to the YaST Control Center.

   g. Close the YaST Control Center by selecting **Quit**.

2. Change the date and time settings by using the YaST graphical user interface:

   During classroom setup, your SLES 9 server was configured to use Mountain Standard Time (US).

To check and change this setting with YaST (GUI), do the following:

a. From the terminal window, logout as root by entering **exit**.

b. Log in as root by entering **sux -**; then enter a password of **novell**.

c. List the available modules for YaST by entering **yast2 -l**.

d. Scroll through the list to find the module you would use to set the time zone.

e. Start the module by entering **yast2 timezone**.

   The Clock and Time Zone Configuration dialog appears.

f. Select ***your region*** and ***your time zone***; then from the Hardware clock drop-down list, select **UTC**.

g. When you finish, select **Accept**.

   Notice that YaST uses SuSEconfig to configure all the necessary files and services for the clock and time zone settings.

   When the configuration is complete, YaST closes. Because you directly accessed the YaST module, the YaST Control Center is not displayed.

3. Check and set the monitor and resolution for your SLES 9 server:

During installation, YaST selects a graphics card driver and resolution automatically.

However, you might want to change these as YaST dialogs (and the terminal window) display best at a 1152 x 768 or a 1024 x 768 resolution.

a. Right-click the desktop and select **Configure Desktop**.

   The Configure - Desktop dialog appears.

b. On the left, select **Size & Orientation**.

c. On the right, from the Screen size drop-down list, select **1152 x 768** or **1024 x 768**.

d. Do one of the following:

- After selecting the screen size, select **OK > Accept Configuration**; then continue on to Step 4.

or

- If either of these settings is not available, continue on to Step e.

e. Start YaST from the desktop by selecting the **YaST** icon.

A Run as root dialog appears.

f. Enter a password of **novell**; then select **OK**.

The YaST Control Center appears.

g. Select **Hardware > Graphics Card and Monitor**.

A Desktop Settings dialog appears with Graphical desktop environment selected and your graphics card and monitor settings listed.

h. Select **Change**.

An SaX2 dialog appears.

i. On the left, expand **Desktop**; then use the **Monitor**, **Graphics card**, and **Color and Resolution** options (with the **Change configuration** button) to make any adjustments to your graphics configuration.

j. When you finish, select **Finalize**; then select **Test**.

k. Do one of the following:

- If the test screen appears properly, select **Save**; then select **OK**.

or

- If the test screen does not display properly (or at all), return to the SaX2 dialog by pressing **Ctrl+Alt+Backspace** and change the Desktop settings.

l. When you finish, select **Accept**.

4. Close the YaST Control Center.

5. (Conditional) If you changed the graphics settings, do the following:

   a. Log out as geeko by selecting **KDE Menu > Logout > Logout**.

      You are returned to the GUI login screen.

   b. Log in again as **geeko** (password of **N0v3ll**) to see the changes you made to the graphics settings.

(End of Exercise)

Objective 3 Monitor Your SLES 9 System

After installation you probably have questions similar to the following;

- Did the system boot normally?
- What is the kernel version?
- What services are running?
- What is the load on the system?

In this objective, you are introduced to the following information that helps you discover information about your hardware and Linux system:

- Boot Log Information (/var/log/boot.msg)
- Hardware Information (/proc/)
- Hardware Information (Command Line Utilities)
- System and Process Information (Command Line Utilities)
- GUI Desktop Utilities

These tools are covered in more detail in later sections of this course, and in other Novell SUSE Linux administration courses.

Boot Log Information (/var/log/boot.msg)

When SLES 9 starts, some lines scroll by too quickly for you to read easily. If there is an error message, it might be nearly impossible to read it.

However, because these messages are kept in a buffer and saved to /var/log/boot.msg, you can view them after booting by entering **dmesg | less**, as in the following:

```
Linux version 2.6.5-7.97-default (geeko@buildhost) (gcc version 3.3.3
(SuSE Linu
x)) #1 Fri Jul 2 14:21:59 UTC 2004
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 00000000000a0000 (usable)
 BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 000000001fdf7000 (usable)
 BIOS-e820: 000000001fdf7000 - 000000001fe16000 (ACPI data)
 BIOS-e820: 000000001fe16000 - 0000000020000000 (reserved)
 BIOS-e820: 00000000fec00000 - 00000000fec10000 (reserved)
 BIOS-e820: 00000000fee00000 - 00000000fee10000 (reserved)
 BIOS-e820: 00000000ffb00000 - 0000000100000000 (reserved)
503MB vmalloc/ioremap area available.
0MB HIGHMEM available.
509MB LOWMEM available.
On node 0 totalpages: 130551
  DMA zone: 4096 pages, LIFO batch:1
  Normal zone: 126455 pages, LIFO batch:16
  HighMem zone: 0 pages, LIFO batch:1
DMI 2.3 present.
ACPI: RSDP (v000 DELL                                      ) @ 0x000fd6d0
ACPI: RSDT (v001 DELL    GX150   0x00000005 ASL  0x00000061) @ 0x000fd6e4
ACPI: FADT (v001 DELL    GX150   0x00000005 ASL  0x00000061) @ 0x000fd718
ACPI: SSDT (v001   DELL    st_ex 0x00001000 MSFT 0x0100000b) @ 0xfffe7465
lines 1-22
```

By using the command less, you can scroll up and down through the messages. The output of dmesg shows messages generated during the initialization of the hardware by the kernel or kernel modules.

For each line displayed at the console during startup, there is one or several lines in the file /var/log/boot.msg.

Although the file /var/log/boot.msg can be somewhat difficult to read, it contains additional information beyond what you can display with dmesg.

This information includes data such as the messages the various scripts generated at boot time and exit status codes, as in the following:

```
/dev/fd0 on /media/floppy type subfs
(rw,nosuid,nodev,sync,fs=floppyfss,procuid)
/dev/hda3 on /apps type ext2 (rw,acl,user_xattr)
/dev/hda5 on /export/data1 type vfat (rw)
/dev/hda6 on /export/data2 type ext3 (rw,usrquota,grpquota)
/dev/hda7 on /export/data3 type reiserfs (rw)
/dev/mapper/project1-pilot on /project1/pilot type reiserfs
(rw,acl,user_xattr)
/dev/mapper/project1-prod on /project1/prod type reiserfs
(rw,acl,user_xattr)
done<notice>exit status of (boot.localfs) is (0)
<notice>run boot scripts (boot.crypto)
<notice>exit status of (boot.crypto) is (0)
<notice>run boot scripts (boot.scpm boot.restore_permissions
boot.loadmodules)
Loading required kernel modules
doneRestore device permissionsdone
<notice>exit status of (boot.scpm boot.restore_permissions
boot.loadmodules) is (0 0 0)
<notice>run boot scripts (boot.swap boot.idedma)
Activating remaining swap-devices in /etc/fstab...
done<notice>exit status of (boot.swap boot.idedma) is (0 0)
<notice>run boot scripts (boot.cycle boot.clock)
```

These additional messages can be useful when troubleshooting.

Besides using the commands dmesg and less from the command line to view the contents of /var/log/boot.msg, you can also use YaST to view the file contents by doing the following:

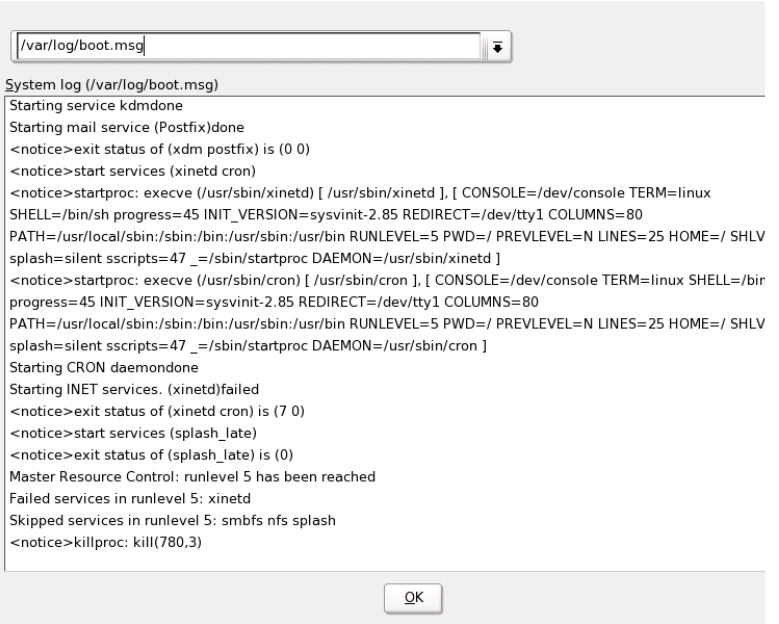
1. From the KDE desktop, start the YaST View System Log module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Misc > View Start-up Log**.

     or

- Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 view_anymsg;** then from the drop-down list, select **/var/log/boot.msg**.

The following appears:

Figure 1-3



2. When you finish reading through the log file, exit by selecting **OK**.

Hardware Information (/proc/)

The directory /proc/ lets you view hardware information stored in the kernel memory space.

For example, if you enter cat /proc/cpuinfo, output is generated from data stored in kernel memory that gives you information such as the CPU model name and cache size.

You can view the available information by using commands such as cat, more, or less with a file name (such as **cat /proc/cpuinfo**).

The following are some of the commonly-used filenames to generate information:

- **/proc/devices.** View the devices used on your Linux system.
- **/proc/cpuinfo.** View processor information.
- **/proc/ioports.** View the I/O ports on your server. The I/O ports are the addresses of various hardware devices.
- **/proc/interrupts.** View the IRQ (hardware interrupt signal) assignments for your Linux system.
- **/proc/dma.** View the DMA (Direct Memory Access) channels used on your Linux system.
- **/proc/bus/pci/devices.** View the PCI (Peripheral Component Interconnect) information on your Linux system.
- **/proc/scsi/scsi.** View a summary of the SCSI (Small Computer System Interface) information on your Linux system.
- **/proc/bus/usb/devices.** View information about the USB (Universal Serial Bus) devices on your Linux system.
- **/proc/bus/usb/drivers.** View information about the USE drivers on your Linux system.

For a list of all the available filenames, enter **ls -al /proc**.

Hardware Information (Command Line Utilities)

The following are utilities you can use from the command line to view information about the hardware on your Linux system:

- **hwinfo.** Entering this command generates and displays a list of specific information about the devices installed on your Linux system.

  To view one page of information at a time, enter **hwinfo | less**. For a summary listing, entering **hwinfo --short**.

  To write the information to a log file enter **hwinfo --log *filename***.

- **hdparm.** Entering this command with various options lets you view information about your hard drive and manage certain hard drive parameters.

  For example, the option -i displays hard drive identification information available at boot time. The option -l requests information directly from the hard drive.

  For a summary list of available options, enter **hdparm** or **hdparm -h**.

- **fdisk.** While this command is primarily used for managing the partition table on a Linux system, you can also use options such as -l (list partition tables), -s (size of partition) to view hard drive information.

- **iostat.** Entering this command displays CPU and input/output (I/O) statistics for devices and partitions.

  This command generates reports that can be used to change system configuration to better balance the input/output load between physical disks.

  The first report generated provides statistics concerning the time since the system was booted. Each subsequent report covers the time since the previous report.

You can generate 2 types of reports with the command—the CPU usage report and the device usage report.

The option -c generates only the CPU usage report; the option -d generates only the device usage report.

- **lspci.** Entering this command displays information about all PCI buses in your Linux system and all devices connected to them.

  The options -v and -vv generate verbose reports. The option -b gives you a bus-centric view of all the IRQ numbers and addresses as seen by the cards (instead of the kernel) on the PCI bus.

- **siga.** This is a SUSE Linux tool for gathering hardware information.

System and Process Information (Command Line Utilities)

The following are commonly-used command line tools for viewing system information, and viewing and managing system processes:

- top
- uptime
- ps
- netstat
- uname

top

The top utility gives you a summary of various system statistics (such as memory and CPU usage, uptime, and number of users) in the top part of the screen, such as the following:

Figure 1-4

```
top - 14:00:36 up 15 days,  2:31,  1 user,  load average: 1.16, 1.15,
Tasks:  95 total,   1 running,  89 sleeping,   5 stopped,   0 zombie
Cpu(s): 97.0% us,  2.6% sy,  0.0% ni,  0.0% id,  0.0% wa,  0.3% hi,  0
Mem:    514712k total,   434520k used,    80192k free,   118432k buffe
Swap:  1028120k total,        8k used,  1028112k free,    82884k cache

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 7182 root      15   0 26276  13m  17m S  3.0  2.7 602:19.49 y2base
16379 root      15   0 85772  17m  68m S  1.3  3.5   1:45.69 X
13676 geeko     16   0  1876  876 1520 S  1.0  0.2   0:34.41 ksysgua
13675 geeko     16   0 26760  15m  23m S  0.7  3.1   1:11.70 ksysgua
16513 geeko     15   0 26212  13m  23m S  0.3  2.8   0:10.33 kdeinit
14127 geeko     15   0 33220  21m  26m S  0.3  4.3   0:15.30 kdeinit
17114 root      16   0  1752  920 1536 R  0.3  0.2   0:00.45 top
    1 root      16   0   588  244  444 S  0.0  0.0   0:05.15 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:00.06 ksoftir
    3 root       5 -10     0    0    0 S  0.0  0.0   0:00.13 events/
    4 root       5 -10     0    0    0 S  0.0  0.0   0:00.00 kacpid
    5 root       5 -10     0    0    0 S  0.0  0.0   0:00.00 kblockd
    6 root       7 -10     0    0    0 S  0.0  0.0   0:00.07 khelper
    7 root      15   0     0    0    0 S  0.0  0.0   0:23.73 pdflush
    8 root      15   0     0    0    0 S  0.0  0.0   0:04.74 pdflush
   10 root      10 -10     0    0    0 S  0.0  0.0   0:00.00 aio/0
    9 root      16   0     0    0    0 S  0.0  0.0   0:12.64 kswapd0
  163 root      23   0     0    0    0 S  0.0  0.0   0:00.00 kseriod
```

In the lower part of the display, processes running on the server are sorted by CPU usage. The screen is updated every 2 seconds.

You can sort the processes by fields such as % Memory (type **m**) or by user (type **u** and then enter the ***user ID***).

You can change the sorting and send signals to a process (such as suspending or killing a process). When you finish, type **q** to end top.

For additional information on top, type **h** while in top or enter **man top** from a command line.

uptime

Although the command top gives you system information in the header, there might be times when you only want specific information without starting a utility.

For example, you can use the command uptime to display the current time, the length of time the system has been running, the number of users on the system, and the average number of jobs in the run queue over the last 1, 5, and 15 minutes.

The following is an example of entering the command uptime:

```
geeko@DA50:~> uptime
  1:44am  up 11 days 18:56,  2 users,  load average: 0.05, 0.09, 0.08
geeko@DA50:~>
```

For additional information on the uptime command, enter **man uptime**.

ps

The utility ps displays the processes running on the system sorted by process ID. You can change the output using one of several options.

For example, to see a display with detailed information, enter **ps aux** (as in the following):

```
geeko@DA50:~> ps aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   588  244 ?        S    Aug20   0:05 init [5]
root         2  0.0  0.0     0    0 ?        SN   Aug20   0:00 [ksoftirqd/0]
root         3  0.0  0.0     0    0 ?        S<   Aug20   0:00 [events/0]
root         4  0.0  0.0     0    0 ?        S<   Aug20   0:00 [kacpid]
root         5  0.0  0.0     0    0 ?        S<   Aug20   0:00 [kblockd/0]
root         6  0.0  0.0     0    0 ?        S<   Aug20   0:00 [khelper]
root         7  0.0  0.0     0    0 ?        S    Aug20   0:00 [pdflush]
root         8  0.0  0.0     0    0 ?        S    Aug20   0:12 [pdflush]
root        10  0.0  0.0     0    0 ?        S<   Aug20   0:00 [aio/0]
root         9  0.0  0.0     0    0 ?        S    Aug20   0:05 [kswapd0]
root       169  0.0  0.0     0    0 ?        S    Aug20   0:00 [kseriod]
root       530  0.0  0.0     0    0 ?        S<   Aug20   0:00 [reiserfs/0]
root       707  0.0  0.0     0    0 ?        S<   Aug20   0:00 [kcopyd]
root       769  0.0  0.0     0    0 ?        S    Aug20   0:00 [kjournald]
root      1426  0.0  0.0     0    0 ?        S    Aug20   0:00 [khubd]
root      1843  0.0  0.1  1584  720 ?        Ss   Aug20   0:27 /sbin/resmgrd
bin       1846  0.0  0.1  1432  624 ?        Ss   Aug20   0:37 /sbin/portmap
```

If there is more information than can fit on one screen, you can use the command less (such as **ps aux | less**).

netstat

While the command ps provides information on a process level, you can use netstat to find out which network ports are offering services and what connections are established, as in the following:

```
geeko@DA50:~> netstat -patune
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
User       Inode      PID/Program name
tcp        0      0 0.0.0.0:2049           0.0.0.0:*              LISTEN
0          1288996    -
tcp        0      0 0.0.0.0:32776          0.0.0.0:*              LISTEN
0          1248510    -
tcp        0      0 0.0.0.0:139            0.0.0.0:*              LISTEN
0          1324989    -
tcp        0      0 10.0.0.50:427          0.0.0.0:*              LISTEN
0          5032       -
tcp        0      0 127.0.0.1:427          0.0.0.0:*              LISTEN
0          5031       -
tcp        0      0 0.0.0.0:111            0.0.0.0:*              LISTEN
0          2844       -
tcp        0      0 0.0.0.0:752            0.0.0.0:*              LISTEN
0          1274848    -
tcp        0      0 0.0.0.0:785            0.0.0.0:*              LISTEN
0          1275487    -
tcp        0      0 0.0.0.0:21             0.0.0.0:*              LISTEN
0          1352119    -
...
```

The following are some useful options for customizing the output of netstat:

Table 1-9

Option	Description
-p	Show processes (as root)
-a	Show listening and non listening sockets (all)
-t	Show tcp information

(continued) **Table 1-9**

Option	Description
-u	Show udp information
-n	Do not resolve hostnames
-e	Display additional information (extend)
-r	Display routing information

uname

You can use the command uname to find out about the current kernel version, as in the following:

```
geeko@DA50:~> uname -a
Linux DA50 2.6.5-7.97-default #1 Fri Jul 2 14:21:59 UTC 2004 i686 i686
i386 GNU/Linux
geeko@DA50:~>
```

GUI Desktop Utilities

The following are utilities you can use from the desktop to view information about your hardware and Linux system:

- KDE System Guard
- Xosview
- SuSEPlugger

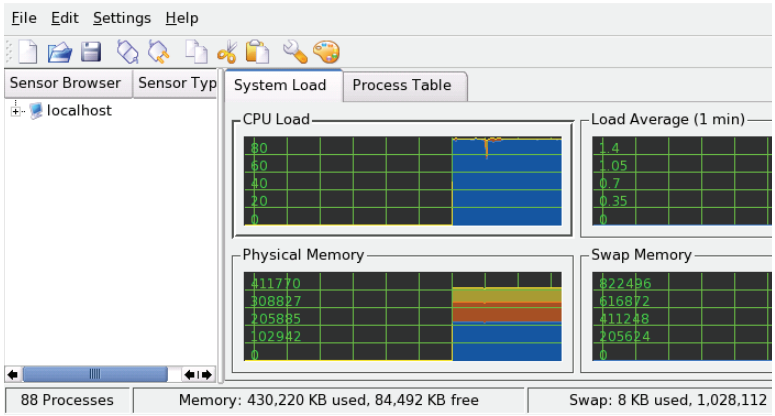
KDE System Guard

KDE System Guard is the KDE desktop task manager and performance monitor. With KDE System Guard, you can monitor system load performance (CPU load, load average, physical memory, and swap memory) or processes.

To access KDE System Guard, from the system menu select **System > Monitor > KDE System Guard**.

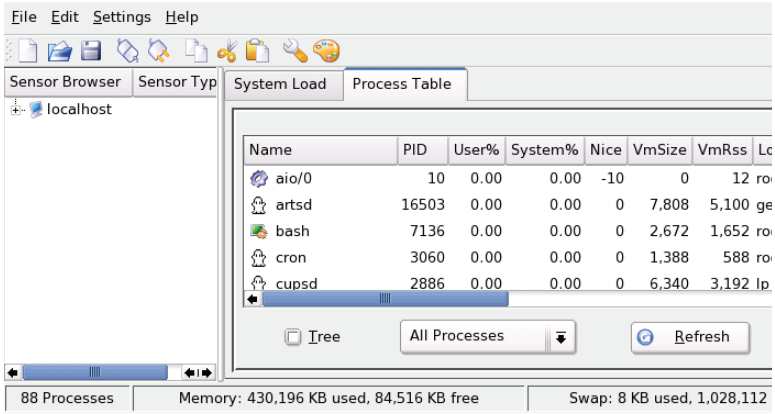
The following appears:

Figure 1-5



Selecting **Process Table** displays the following:

Figure 1-6



From the process table, you can monitor and stop (kill) processes.

For additional information on KDE System Guard, from the program menu select **Help > KDE System Guard Handbook**.

Xosview

You can use the utility Xosview to display the status of several system-based parameters such as CPU usage, load average, memory usage, swap space usage, network usage, interrupts, and serial port status.

To start Xosview, do one of the following:

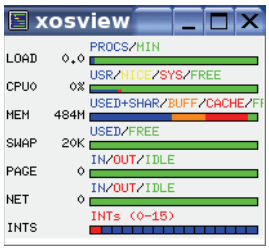
- From the KDE menu, select **System > Monitor > Xosview**.

  or

- From a terminal window, enter **xosview**.

A window similar to the following appears:

Figure 1-7



Each parameter status is displayed as a horizontal bar separated into color-coded regions. Each region represents a percentage of the resource that is being put to a particular use.

When you finish viewing the information, you can quit by closing the window or by typing **q**.

SuSEPlugger

You can use the utility SuSEPlugger to view details about specific hardware on your Linux system, and to launch the appropriate YaST module for configuring the hardware.

To start SuSEPlugger, do one of the following:

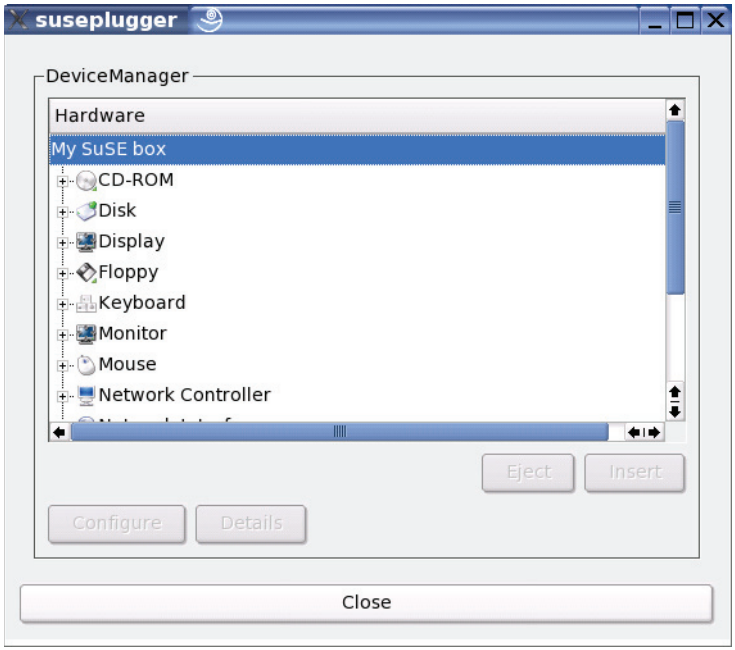
- From your desktop panel (bottom of the screen), select the **SuSEPlugger** icon.

  or

- From a terminal window, enter **suseplugger**.

The following appears:

Figure 1-8



From the SuSEPlugger window you can expand a hardware category and select a specific hardware device.

After selecting the hardware device, you can view details about the device by selecting **Details**, or launch a YaST module to configure the device by selecting **Configure**.

When you finish using SuSEPlugger, close the window by selecting **Close**.

Exercise 1-3 Gather Information About Your SLES 9 Server

After installing SLES 9, you decide to use several of the administration tools available to gather information about your SLES 9 server.

Enter information in the following table from steps 1-10:

Table 1-10

System Parameter	Value
OS	
Hardware Architecture	
Processor Type	
Hostname	
Kernel Release	
Kernel Version (include date and time)	
System Up Time	
Load Averages	
SLES 9 Version	
System Date and Time	
Model Name of Processor	
Free Memory	

Do the following:

1. From the KDE desktop (logged in as geeko), open a terminal window.

2. View the kernel release of the SLES 9 distribution you are running by entering **uname -r**.
3. View the computer (machine) hardware architecture by entering **uname -m**.
4. View the processor type for this Linux build by entering **uname -p**.
5. View all information including hostname, kernel release, and kernel version by entering **uname -a**.
6. View how long the system has been running and the load averages by entering **uptime**.
7. View the version of the SLES 9 distribution by entering **cat /etc/SuSE-release**.
8. View the system date and time by entering **date**.
9. View information about the current running processor by entering **cat /proc/cpuinfo**.
10. View the current memory statistics by entering **cat /proc/meminfo**.
11. Start Xosview to monitor system resource usage:
    a. Press **Alt+F2**.

       The Run Command - KDesktop dialog appears.

    b. Enter **xosview**; then select **Run**.
12. Use SuSEPlugger to view unknown devices:
    a. From the System Tray, select the **SuSEPlugger** icon (SUSE Hardware Tool).

       SuSEPlugger is similar to the Device Manager in Windows.

    b. View the unconfigured devices on your SLES 9 server by expanding the **Unknown** category.

13. Launch the YaST Network Card module from SuSEPlugger:
    a. Expand **Network Controller**; then select ***your network device***.
    b. Select **Configure**.

       The Run as root dialog appears.
    c. Enter a password of **novell**; then select **OK**.

       The Network cards configuration dialog appears.
    d. Close the dialog without changing any settings by selecting **Abort**; then select **Yes**.
14. Close all open windows on your desktop.

(End of Exercise)

Summary

Objective	Summary
1. Describe SLES 9	Backed by an extensive Novell support infrastructure and partner network, SLES 9 is a secure, reliable platform for open source computing in the enterprise. SLES 9 also provides open application programming interfaces (APIs) and other development tools that simplify Linux* integration and customization. In this objective you learned the following about SLES 9: ■ What's New in SLES 9 ■ Hardware Support and System Requirements ■ Service and Support ■ Linux Standards and SLES 9
2. Access and Use YaST	YaST stands for Yet another Setup Tool. You can use YaST to complete many configuration tasks as a SUSE Linux Enterprise Server administrator. To effectively use YaST for configuring SLES 9, you need to know the following: ■ YaST Basics ■ The Role of SuSEconfig

Objective	Summary
3. Monitor Your SLES 9 System	After installation you probably have questions similar to the following; ■ Did the system boot normally? ■ What is the kernel version? ■ What services are running? ■ What is the load on the system? In this objective, you were introduced to the following information that helps you discover information about your hardware and Linux system: ■ Boot Log Information (/var/log/boot.msg) ■ Hardware Information (/proc/) ■ Hardware Information (Command Line Utilities) ■ System and Process Information (Command Line Utilities) ■ GUI Desktop Utilities

SECTION 2 Manage User Access and Security

In this section you learn how to perform basic user and group management tasks that provide users with a secure and accessible SUSE Linux Enterprise Server environment.

Objectives

1. Describe Basic Linux User Security Features
2. Manage Linux Users and Groups
3. Manage and Secure the Linux User Environment
4. Secure Files and Directories With Permissions
5. Configure User Authentication With PAM
6. Implement and Monitor Enterprise Security Policies

Objective 1 Describe Basic Linux User Security Features

One of the main characteristics of a Linux operating system is its ability to handle several users at the same time (multiuser) and to allow these users to perform several tasks on the same computer simultaneously (multitask).

To maintain an environment where data and applications are secure, you need to understand the following:

- File System Security Components
- Users and Groups
- Ownership and Access Permissions

File System Security Components

As with other operating systems, you control access to files in a Linux file system by implementing the following types of components:

- **Users.** Users are individual accounts on the Linux system.
- **Groups.** Groups are collections of users. Users are assigned to a group when they are created. Only root or the owner can change the group to which the file or directory is assigned. Every user must belong to at least 1 group.
- **Ownership.** The user who creates a file or directory is automatically assigned as its owner. Ownership can only be changed manually by root.
- **Permissions.** Permissions determine user access to a file or directory.

Users and Groups

Because Linux is a multiuser system, several users can work on the system at the same time. For this reason the system uniquely identifies all users through user accounts that require a user name and password to log in to the system.

In addition, Linux provides groups that let you associate users together that require the same type of access privileges to data and applications.

To manage users and groups, you need to know the following:

- User and Group ID Numbers
- Regular vs. System Users
- Public vs. Private Group Schemes
- User Accounts and Home Directories
- User and Group Configuration Files
- How to Check /etc/passwd and /etc/shadow

User and Group ID Numbers

Because an operating system can handle numbers much better than strings, users and groups are administered internally as numbers on a Linux system.

The number which a user receives is the called a *user ID* (UID). Every Linux system has a privileged user, the user root. This user always has a UID of 0. UID numbering for normal users starts (by default) at 1000 for SUSE Linux.

As with users, groups are also allocated a number internally called the *group ID* (GID).

Normal users are usually included in the group users. Other groups also exist (and can be created) for special roles or tasks.

For example, all users who intend to create web pages can be placed in the group webedit. Of course, file permissions for the directory in which the web pages are located must be set so that members of the group webedit are able to write and read files.

You can use the command id to display information about a user's UID and which groups she is assigned. For example, entering **id geeko** provides information about the user geeko:

```
geeko@earth:~> id geeko
uid=1000(geeko) gid=100(users)
groups=100(users),14(uucp),16(dialout,17(audio),
33(video)
```

This information includes the following:

- **User ID:** uid=1000(geeko)
- **Current default (effective) group:** gid=100(users)
- **All groups of which geeko is a member:** groups=100(users),14(uucp),16(dialout),...

If you want information on the groups in which you are a member, enter **groups**. You can specify a particular user by entering **groups *user***.

You can display additional information about local users by entering **finger *user***, as illustrated in the following:

```
geeko@earth:~> finger geeko
Login: geeko                              Name: geeko
Directory: /home/geeko                    Shell: /bin/bash
On since Thu Oct 23 13:21 (CEST) on pts/0 from 192.168.5.16
New mail received Wed Oct 22 11:54 2003 (CEST)
          Unread since Wed Oct 22 11:54 2003 (CEST)
No Plan.
geeko@earth:~ >
```

Regular vs. System Users

In a Linux operating system, there are 2 basic kinds of user accounts:

- **Regular (normal) users.** These are user accounts you create that allow employees and others to log in to the Linux environment. This type of login gives people a secure environment for accessing data and applications.

  These user accounts are managed by the system administrator.

- **System users.** These are user accounts created during installation that are used by services, utilities, and other applications to run effectively on the server.

Regular users are stored in the files /etc/passwd and /etc/shadow; system users are created by scripts that are part of rpm packages.

Public vs. Private Group Schemes

When you create a user in a Linux (or UNIX) environment, that user is assigned a default group using one of 2 basic methods (schemes):

- **Private scheme.** In this scheme, the user is assigned his own group that he can manage.

  For example, if you create the user **cgrayson**, a group **cgrayson** is also created.

- **Public scheme.** In this scheme, the user is assigned to a general, public group such as **users**.

  Because the group includes all new users, the group is normally managed by the system administrator.

SUSE Linux Enterprise Server uses the public scheme for assigning new users to a group.

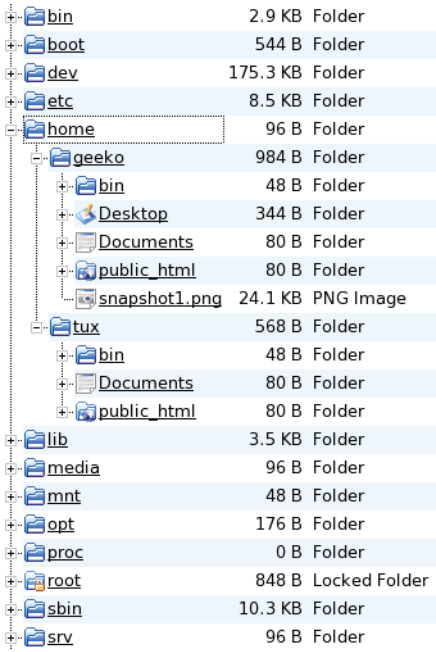
User Accounts and Home Directories

Each user has a user account identified by a login name and a personal password for logging in to the system.

By having user accounts, you are able to protect a user's personal data from being modified, viewed, or tampered with by other users. Each user can set up her own working environment and always find it unchanged when she logs back in.

As part of these security measures, each user in the system has her own directory in the directory /home/, as shown in the following:

Figure 2-1



The exception to this rule is the account root. It has its own home directory in /root/.

Home directories allow personal data and desktop settings to be secured for user access only.

You should avoid using the root account when performing day-to-day tasks that do not involve system management.

User and Group Configuration Files

The Linux system stores all user and group configuration data in the following files:

- /etc/passwd
- /etc/shadow
- /etc/group

Whenever possible, you should not modify these files with an editor. Instead use the Security and Users modules provided in YaST or the command line tools described in "Manage User Accounts From the Command Line" on 2-28.

Modifying these files with an editor can lead to errors (especially in /etc/shadow), such as a user—including the user root—no longer being able to log in.

/etc/passwd

The file /etc/passwd stores information for each user such as the user name, the UID, the home directory, and the standard shell.

In the past, /etc/passwd also contained the encrypted password. However, because the file needs to be readable by all (especially to find out the UID of a particular user), the encrypted password is now stored in /etc/shadow, which is only readable by root.

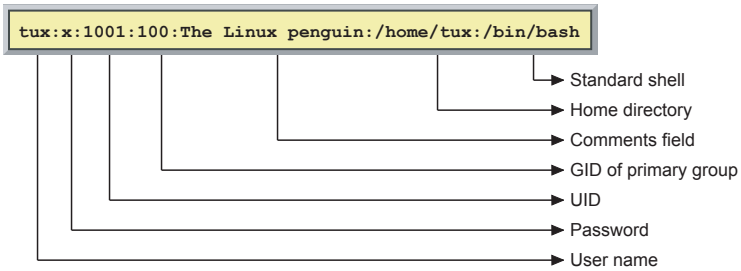
The following is a sample /etc/passwd file:

Figure 2-2

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
news:x:9:13:News system:/etc/news:/bin/bash
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
games:x:12:100:Games account:/var/games:/bin/bash
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
mdom:x:28:28:Mailing list agent:/usr/lib/majordomo:/bin/bash
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
squid:x:31:65534:WWW-proxy squid:/var/cache/squid:/bin/false
amanda:x:37:6:Amanda admin:/var/lib/amanda:/bin/bash
irc:x:39:65534:IRC daemon:/usr/lib/ircd:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
named:x:44:44:Name server daemon:/var/lib/named:/bin/false
gdm:x:50:15:Gnome Display Manager daemon:/var/lib/gdm:/bin/bash
geeko:x:1000:100:geeko:/home/geeko:/bin/bash
tux:x:1001:100:The Linux Penguin:/home/tux:/bin/bash
```

Each line in the file /etc/password represents 1 user, and contains the following information:

Figure 2-3



Note the following about the fields in each line:

- **User name.** This is the name a user enters to log in to the system (login name).

  Although Linux can handle longer user names, in this file they should be restricted to a maximum of 8 characters for backward compatibility with older programs.

- **Password.** The x in this field means that the password is stored in the file /etc/shadow.
- **UID.** In compliance with the Linux standards, there are 2 number ranges which are reserved:
  - **0–99** for the system itself
  - **100–499** for special system users (such as services and programs)

  Normal users start from UID 1000.
- **Comments field.** Normally, the full name of the user is stored here. Information such as a room number or telephone number can also be stored here.
- **Home directory.** The personal directory of a user is normally in the directory /home/ and is the same name as the user (login) name.
- **Standard shell.** This is the shell that is started for a user after he or she has successfully logged in. In Linux this is normally bash (Bourne Again Shell).

  The shell must be listed in the file **/etc/shells**. Each user can change his standard shell with the command chsh (see **man chsh**).

For additional information on this file, enter **man 5 passwd**.

/etc/shadow

The /etc/shadow file stores encrypted user passwords and password expiration information. Most Linux systems use *shadow passwords*. Shadow passwords are stored in /etc/shadow instead of /etc/passwd.

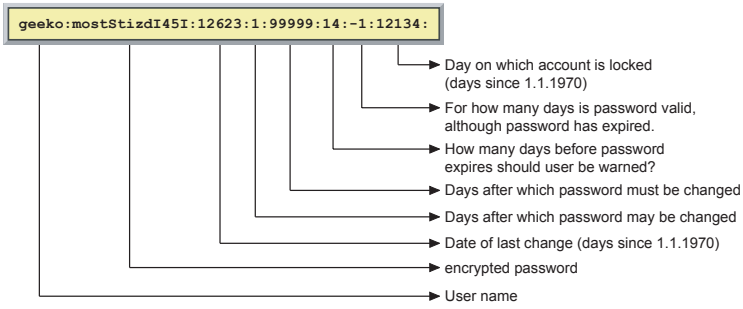
The file can only be changed by the user root and read by the user root and members of the group shadow. The following is a sample /etc/shadow file:

Figure 2-4

```
mailman:!:12608:0:99999:7:::
man:*:8902:0:10000::::
mdom:!:12 08:0:99999:7:::
mysql:!:12608:0:99999:7:::
named:!:12608:0:99999:7:::
news:*:8902:0:10000::::
nobody:*:8902:0:10000::::
ntp:!:12608:0:99999:7:::
pop:!:12608:0:99999:7:::
postfix:!:12608:0:99999:7:::
postgres:!:12608:0:99999:7:::
quagga:!:12608:0:99999:7:::
radiusd:!:12608:0:99999:7:::
root:XOQeyibhsgHj2:12608:0:10000::::
snort:!:12608:0:99999:7:::
squid:!:12608:0:99999:7:::
sshd:!:12608:0:99999:7:::
stunnel:!:12608:0:99999:7:::
uucp:*:8902:0:10000::::
vscan:!:12608:0:99999:7:::
wwwrun:*:8902:0:10000::::
tux:svSIYQsFoEwKg:12608:0:99999:7:-1::
geeko:mostSt1zdI45I:12623:1:99999:14:-1:12134:
```

Each line in the file /etc/shadow belongs to 1 user and contains the following fields:

Figure 2-5



The above illustration shows the entry for the user **geeko** with an encrypted password. The plain text password was **suse**.

The encrypted password is coded with the crypt function and is always 13 characters in length. The encrypted word consists of letters, digits, and the special characters “.” and “/.”

If an invalid character occurs in the password field (such as “*” or “!”), then that user has an invalid password.

Many users, such as wwwrun (Apache Web server) or bin have an asterisk (“*”) in the password field. This means that these users do not log in to the system, but instead play a role for specific programs.

If the password field is empty, then the user can log in to the system without entering a password. A password should always be set in a multiuser system.

/etc/group

The file /etc/group stores group information. The following is a sample /etc/group file:

Figure 2-6

```
root:x:0:
bin:x:1:daemon
daemon:x:2:
sys:x:3:
tty:x:5:
disk:x:6:
lp:x:7:
www:x:8:
kmem:x:9:
uucp:x:14:geeko,tux
shadow:x:15:
dialout:x:16:geeko,tux
audio:x:17:geeko,tux
floppy:x:19:
cdrom:x:20:
console:x:21:
utmp:x:22:
at:!:25:
postgres:!:26:
mdom:!:28:
public:x:32:
video:x:33:geeko,tux
nobody:x:65533:
nogroup:x:65534:nobody
users:x:100:
novell:!:1000:
```

Each line in the file represents a single group record, and contains the group name, the GID (group ID) and the members of the group. For example

```
video:x:33:geeko,tux
```

This is the entry for the group video in /etc/group and has a GID of **33**. The users geeko and tux are members of this group. The second field (**x**) is the password field.

The /etc/groups file shows secondary group memberships, but does not identify the primary group for a user.

In older versions of SUSE Linux (such as SUSE Linux Enterprise Server 8), group passwords are stored in the file /etc/gshadow.

How to Check /etc/passwd and /etc/shadow

Because user configuration is handled with 2 files (/etc/passwd and /etc/shadow), these files match each other. This includes an entry for each user in both files.

However, discrepancies can occur—especially if you are configuring these files in an editor. In these cases, there are programs you can use to check /etc/passwd and /etc/shadow.

For example, to view the contents of both files at once, you can enter the following:

```
earth:~ # tail -3 /etc/passwd /etc/shadow
==> /etc/passwd <==
cyrus:x:96:12:User for cyrus-imapd:/usr/lib/cyrus:/bin/bash
tux:x:1000:100:tux:/home/tux:/bin/bash
geeko:x:1001:100:geeko:/home/geeko:/bin/bash
==> /etc/shadow <==
postfix:!:12543:0:99999:7:::
cyrus:!:12543:0:99999:7:::
tux:0C9zaAMz3p72g:12551:0:99999:7:::
earth:~ #
```

In the above example, the user geeko is entered in /etc/passwd but not in /etc/shadow.

In order to correct this type of error, you can enter the command pwconv:

```
earth:~ # pwconv
earth:~ # tail -3 /etc/passwd /etc/shadow
==> /etc/passwd <==
cyrus:x:96:12:User for cyrus-imapd:/usr/lib/cyrus:/bin/bash
tux:x:1000:100:tux:/home/tux:/bin/bash
geeko:x:1001:100:geeko:/home/geeko:/bin/bash
==> /etc/shadow <==
cyrus:!:12543:0:99999:7:::
tux:0C9zaAMz3p72g:12551:0:99999:7:::
geeko:x:12566:0:99999:7:::0
earth:~ #
```

You can also use the command pwck:

```
earth:~ # pwck
Checking '/etc/passwd'
User 'geeko': directory '/home/geeko' does not exist.
Checking '/etc/shadow'.
earth:~ #
```

Exercise 2-1 Check User and Group Information on Your Server

Check the user and group information on your SLES 9 server by doing the following from a command line:

1. Make sure you are logged in as **geeko** to the KDE desktop.
2. From a terminal window, su to root (**su -**) with a password of **novell**.
3. Display all information in the file /etc/group by entering **cat /etc/group**.
4. Display only the group name and group number fields in the file by entering **cat /etc/group | cut -d: -f1,3 | less**.

   You see entries similar to root:0 and bin:1.
5. Scroll through the file by pressing the **spacebar**; then exit by typing **q**.
6. Display the contents of the file /etc/passwd by entering **less /etc/passwd**.
7. Scroll through the file by pressing the **spacebar**; then exit by typing **q**.
8. Display the identity information of the logged in user by entering **id**.

   Because you are su'd to root, you see UID, GID, and group information for root.
9. Exit the su state and return to the geeko user by entering **exit**.
10. Enter **id** again.

    Notice that the groups displayed for geeko are different from those displayed for root.
11. Close the terminal window by entering **exit**.

(End of Exercise)

Ownership and Access Permissions

Each file and directory in the file system is assigned access permissions. The permissions assigned determine the level of access a given user has. Permissions are assigned at 3 levels:

- **Owner.** The permissions assigned to a file or directory's owner determine the owner's level of access.
- **Group.** Permissions assigned to the group determine the level of access group members have to the file or directory.
- **Others.** Permissions assigned to this entity apply to authenticated users who are not members of the group that has been associated with the file or directory.

For details on assigning permissions, see "How to Set Permissions From the Command Line" on 2-69.

Objective 2 Manage Linux Users and Groups

To manage Linux user accounts groups from your SUSE Linux Enterprise Server, you need to know how to do the following:

- Create and Edit User Accounts With YaST
- Create and Edit Groups With YaST
- Edit User Account Properties
- Configure Account Password Settings
- Manage User Accounts From the Command Line
- Manage Groups From the Command Line
- Create Text Login Messages

Create and Edit User Accounts With YaST

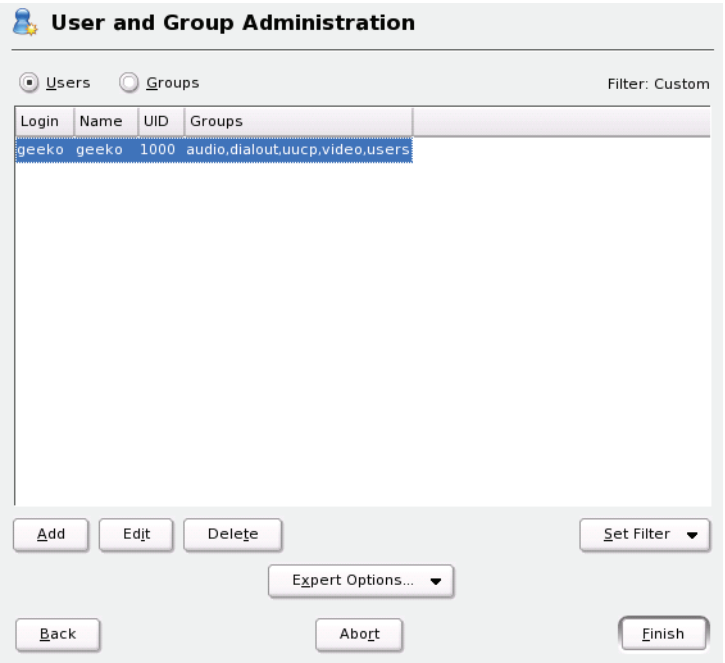
You can use the Edit and Create Users module in YaST to create, edit, and delete Linux user accounts by doing the following:

1. From the KDE desktop, start the YaST Edit and create users module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Security and Users > Edit and create users**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 users**.

The following appears:

Figure 2-7



A list of users (accounts on your server) appears with information such as login name, full name, UID, and associated groups included for each user.

2. Select **Set Filter**; then select one of the following to change the users listed:
   - **Local Users.** User accounts you have created on your local server for logging into the server.
   - **System Users.** User accounts created by the system for use with services and applications.
   - **Custom.** A customized view of users based on the settings configured with **Customize Filter**.

- **Customize Filter.** This option lets you combine listed user sets (such as **Local Users** and **System Users**) to display a customized view (with **Custom**) of the users list.

Additional sets of users (such as LDAP users) are added to the Set Filter drop-down list as you configure and start services on your server.

3. You can create a new user account or edit an existing account by selecting **Add** or **Edit**.

   The following appears:

Figure 2-8

4. Enter or edit information in the following fields:
   - **Full User Name.** Enter a descriptive user name (such as **Sandy Geeko**)

- **User Login.** Enter a user name that can be used to log in to the system (such as **geeko**).
- **Password** and **Verify Password.** Enter and re-enter a password for the user account.

  When entering a password, distinguish between uppercase and lowercase letters.

  Valid password characters include letters, digits, blanks, and #*,.;:._-+!$%&/|?{[()]}=.

  The password should not contain any special characters (such as accented characters), as you might find it difficult to type these characters on a different keyboard layout when logging in from another country.

  With the current password encryption (DES), the password length should be between 5 and 8 characters.

For information on creating longer passwords, see the Password Settings configuration settings under "Configure Security Settings" on 2-54.

5. Save the settings for the new or edited user by selecting **Create** or **Next**.

   The new user appears in the list.

6. Configure your server with the new settings by selecting **Finish**.

Create and Edit Groups With YaST

You can use the Edit and Create Groups module in YaST to create, edit, and delete Linux groups by doing the following:

1. From the KDE desktop, start the YaST Edit and create groups module by doing one of the following:

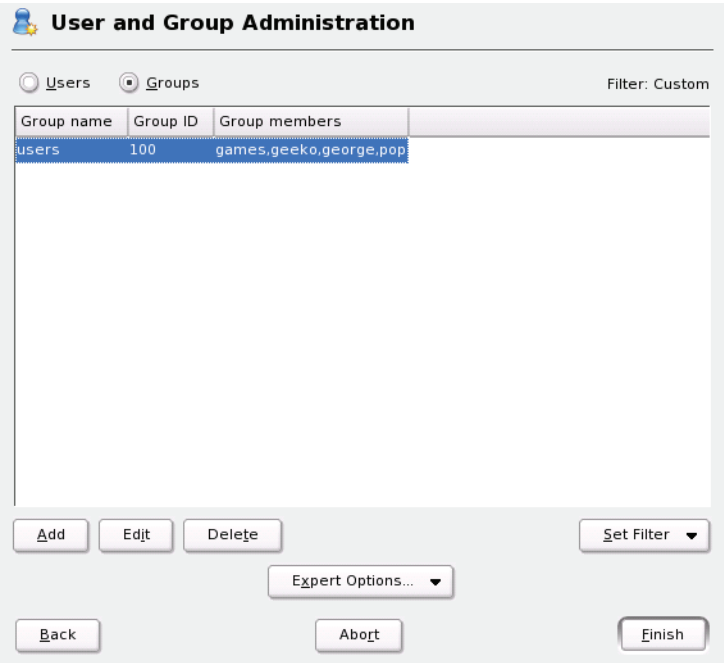
- Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Security and Users > Edit and create groups**.

  or

- Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 groups**.

The following appears:

Figure 2-9



A list of groups appears with information such as group name, Group ID (GID), and group members.

2. Select **Set Filter**; then select one of the following to change the groups listed:

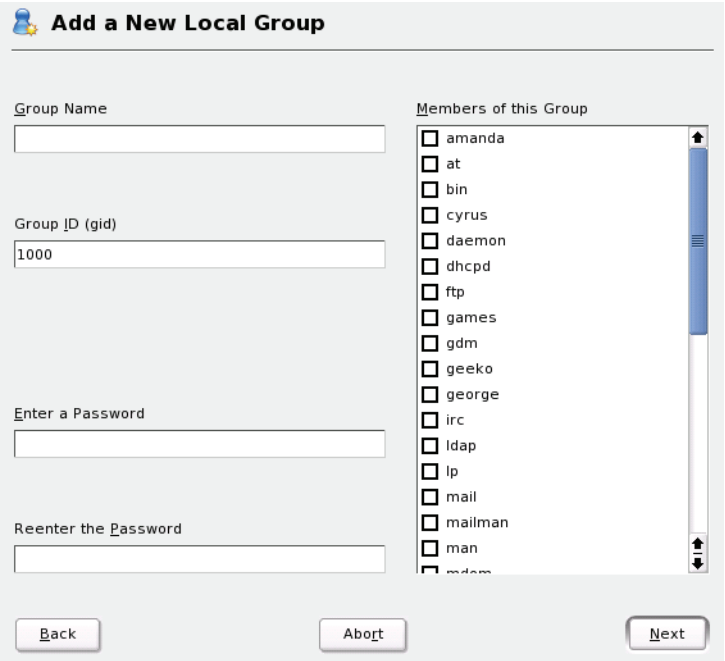
- **Local Groups.** Groups created on your local server to provide permissions for members assigned to the group.
- **System Groups.** Groups created by the system for use with services and applications.
- **Custom.** A customized view of groups based on the settings configured with **Customize Filter**.
- **Customize Filter.** This option lets you combine listed group sets (such as **Local Groups** and **System Groups**) to display a customized view (with **Custom**) of the groups list

Additional sets of groups are added to the Set Filter drop-down list (such as LDAP) as you configure and start services on your server.

3. You can create a new group or edit an existing group by selecting **Add** or **Edit**.

The following appears when you select Add:

Figure 2-10



4. Enter or edit information in the following fields:
   - **Group Name.** The name of the group. Avoid long names. Normal name lengths are between 2 and 8 characters.
   - **Group ID (gid).** The GID number assigned to the group. The number must be a value between 0 and 60000. YaST warns you if you try to use a GID that is already in use.
   - **Enter a Password** (optional)**.** Require the members of the group to identify themselves while switching to this group (see **man newgrp**). To do this, assign a password.

     For security reasons, the password is represented by asterisks (*).

- **Reenter Password.** Enter the password a second time to avoid typing errors.
- **Members of This Group.** Select which users should be members of this group.

  A second list appears (when you select Edit) that shows users for which this group is the default group. This list cannot be edited from YaST.

5. When you finish entering or editing the group information, select **Next**.

   You are returned to the Group Administration dialog.

6. Save the configuration settings by selecting **Finish**.

Edit User Account Properties

You can use YaST to edit user account properties (such as UID or home directory) by doing the following:

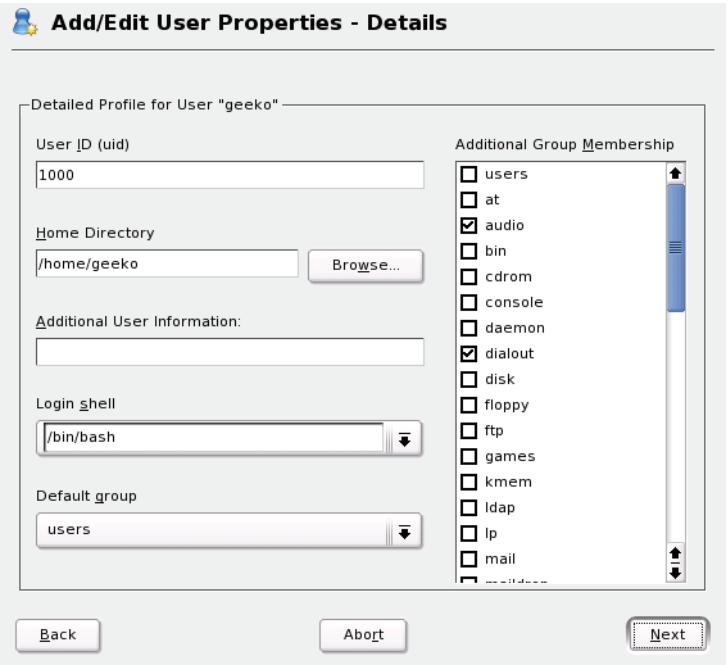
1. From the KDE desktop, start the YaST Edit and create users module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Security and Users > Edit and create users**.

     or

   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 users**.
2. From the user list, select the ***user account*** you want to modify; then select **Edit**.
3. Edit the user account properties by selecting **Details**.

The following appears:

Figure 2-11



4. Enter or edit information in the following fields:

   - **User ID (uid).** For normal users, you should use a UID greater than 499 because the smaller UIDs are used by the system for special purposes and pseudo logins.

     If you change the UID of an existing user, the rights of the files this user owns must be changed. This is done automatically for the files in the user's home directory, but not for files located elsewhere.

If this does not happen automatically, you can change the rights of the user files in the home directory (as root) by entering **chown -R *username* /home/*username***.

- **Home Directory.** The home directory of the user. Normally this is **/home/*username***.

  You can select an existing directory by selecting **Browse**.

- **Additional User Information.** This field can contain up to 3 parts separated by commas. It is often used to enter ***office,work phone,home phone***.

  This information is displayed when you use the **finger** command on this user.

- **Login shell.** From the drop-down list select the default login shell (command interpreter) for this user from the shells installed on your system.

- **Default Group.** This is the group to which the user belongs. Select a group from the list of all groups configured on your system.

- **Additional group membership.** Select all additional memberships you want to assign to the user.

5. When you finish configuring the user account properties, continue by selecting **Next**.

6. Save the configuration settings by selecting **Next > Finish**.

Configure Account Password Settings

You can use YaST to configure password settings (such as expiration date) for individual user accounts by doing the following:

1. From the KDE desktop, start the YaST Edit and create users module by doing one of the following:

   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Security and Users > Edit and create users**.

     or

   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 users**.

2. From the user list, select a ***user account***; then select **Edit**.
3. Select Password Settings.

   The following appears:

Figure 2-12

4. Enter or edit information in the following fields:
   - **Days before Password Expiration to Issue Warning.** Enter the number of days before password expiration that a warning is issued to users.

     Enter **-1** to disable the warning.
   - **Days after Password Expires with Usable Login.** Enter the number of days after the password expires that users can continue to log in.

     Enter **-1** for unlimited access.

- **Maximum number of days for the same password:** Enter the number of days a user can use the same password before it expires.
- **Minimum number of days for the same password:** Enter the minimum age of a password before a user can change it.
- **Expiration date:** Enter the date when the account expires. The date must be in the format YYYY-MM-DD.

  Leave the field empty if the account never expires.

5. Save the password configuration settings by selecting **Next > Next > Finish**.

Manage User Accounts From the Command Line

When logged into the command line as the root user, you can use the following commands to perform the same user management tasks available with YaST (and some tasks not available with YaST):

- **useradd.** You can create a new user account with the useradd command. The following are examples of using the command useradd:
  - Create a bwayne user account (with default parameters):

    ```
    useradd bwayne
    ```

  - Create a bwayne user account (with default parameters) and a home directory (/home/bwayne/):

    ```
    useradd -m bwayne
    ```

  - Create a user account with a specific group membership, and that expires at the end of 2006:

    ```
    useradd -g temp -e 2006-12-31 bwayne
    ```

In addition to the useradd command, you can use the passwd command to change the password for a user account, as in the following:

```
passwd bwayne
```

You are asked to enter the password twice. When you finish, a Password changed message appears.

- **userdel.** This command lets you delete an existing user account. It provides a single option **-r**, which deletes the user's home directory and the user's account.

  Before using **userdel -r**, it is important that you determine the user's UID (**id user**). The UID enables you to locate files outside the user's home directory that are assigned to the user (such as **/var/mail/$USER**).

  To delete these files enter the command:

  ```
  find / -uid user_UID -exec rm {} \;
  ```

- **usermod.** This command lets you modify settings (such as UID, standard shell, home directory, and primary group) for an existing user account.

  The usermod options are basically the same as those for the useradd command.

  The following are examples:

  - Change the home directory:

    ```
    usermod -d /newhome/geeko -m geeko
    ```

  - Change the UID:

    ```
    usermod -u 1001 geeko
    ```

- **passwd.** This command lets you change a user's password.

  When logged in, any user can change his password by entering **passwd** without options; root can change the password of any user by entering **passwd *username***.

Besides changing a user's password, you can also use the command to do the following:

- **Lock a user account.** With the option -l (lock), you can deactivate a user account, and then reactivate the account with the option *-u* (unlock),

  For example, to deactivate the user account geeko, enter **passwd -l geeko**.

- **Display the password status of a user account.** The option -S lets you display the status of a user account. For example, entering **passwd -S geeko** might display the following:

  ```
  geeko L 09/04/2001 0 99999 7 0
  ```

  The status follows directly after the username. **L** means that the user is locked out. Other options are NP (no password) or P (valid password).

  This is followed by the date of the last password change, the minimum length of validity, the maximum length of validity, and the warning periods and inactivity periods when a password expires.

- **Change password times.** You can use options such as -n and -w to change expiration times for user passwords.

  For example, entering **passwd -x 30 -w 5 geeko** changes the maximum number of days to **30** for which the password is valid and warns the user **5** days in advance of the password expiration.

You can learn more about these commands by referring to the online manual pages (such as **man useradd**).

Manage Groups From the Command Line

You can use the following commands to perform the same group management tasks available with YaST (and some tasks not available with YaST):

You need to be logged in as root (or switch to root by entering **su -**) to use these commands.

- **groupadd.** You can create a new group by entering **groupadd *group_name***. In this case, the next free GID is used.

  Using the option -g (such as **groupadd -g 200 sports**) lets you specify a GID.

  Using the option -p (such as **groupadd -p novell sports**) lets you specify a password. You can use the command mkpasswd to create the encrypted password.

  You can verify that the group has been added to the system by entering **tail /etc/group**.

- **groupdel.** You can delete a group by entering **groupdel *group_name***. There are no options for this command.

  You can only delete a group if no user has this group assigned as a primary group.

- **groupmod.** You can modify the settings (such as GID, group name, and users) for an existing group.

  The following are examples:

  - Change the GID:

    ```
    groupmod -g 201 sports
    ```

  - Change the group name from sports to water:

    ```
    groupmod -n water sports
    ```

  - Add the user tux to the group:

    ```
    groupmod -A tux sports
    ```

You can learn more about these commands by referring to the online manual pages (such as **man groupadd**) or online help page (such as **groupadd --help**).

Create Text Login Messages

You can create text login messages that are useful for displaying information when a user logs in from a terminal window, a virtual terminal, or remotely (such as an ssh login).

You can modify the following files to provide these messages:

- **/etc/issue.** Edit this file to configure an initial message for users logging into the system.

  The following is an example of an edited /etc/issue file:

```
Welcome to SUSE Linux Enterprise Server 9 (i586) Kernel \r (\l).

================================================================
                The SUSE Linux Web Server
================================================================
```

- **/etc/motd.** Edit this file to configure an initial message of the day.

  The following is an example of an edited /etc/issue file:

```
This server is currently being maintained by the Web Master at
webmstr@enterprise.com. If you would like to know how to upload files to
this server, contact Human Resources for training.

Recent additions to this server include web portals for Finance and
Customer Relations.
```

Make sure you add one or two empty lines at the end of the message, or it will run into the command line prompt.

Exercise 2-2 Create and Manage Users and Groups From the Command Line

You need to set up your SLES 9 server with user accounts and groups to help train the database administrators in your Digital Airlines office.

Do the following:

- Part I: Customize User Account Default Settings
- Part II: Create Login Messages
- Part III: Create Users With YaST
- Part IV: Create a User Account at the Command Line
- Part V: Manage Groups With YaST
- Part VI: Manage Groups From the Command Line
- Part VII: Manage User Accounts From the Command Line

Part I: Customize User Account Default Settings

In this part of the exercise, you customize the user environment for a new project for database administrators in your Digital Airlines office.

1. From the KDE desktop, open a terminal window and su to root (**su -**) with a password of **novell**.
2. Create a new group named dba by entering the following:

   groupadd dba
3. Edit the file /etc/default/useradd to change the default location for home directories and the list of groups:
   a. From the KDE Menu, select **System > File Manager > File Manager - Super User Mode**.
   b. Enter the password **novell**; then select **OK**.

c. In the Location field, enter **/etc/default**.

d. Right-click **useradd**; then select **Open With > Kate**.

The file useradd opens in the Kate editor.

e. Change the HOME= parameter to the following setting:

HOME=**/export/home**

f. Add dba to the GROUPS= parameter:

GROUPS=dialout,uucp,video,audio,**dba**

g. Save the changes by selecting **File > Save**; then close the file by selecting **File > Close** (keep the Kate window open).

h. From the terminal window, verify the change by entering **cat /etc/default/useradd**.

The database administrator accounts need to have dba and .dba directories created when a new account is created. You can do this by creating the directories in /etc/skel.

4. To create these directories in the directory /etc/skel, enter the following commands:

cd /etc/skel
mkdir dba
mkdir .dba

5. Verify that the directories exist by entering **ls -al**.

6. Create an /export/data2/db directory for storing project files by entering **mkdir -p /export/data2/db**.

7. Verify that the directory exists by entering **ls -al /export/data2**.

Part II: Create Login Messages

Because your server will be used as a database server, you decide to set the initial message for the database administrators at system login to include a message.

1. Edit the file /etc/issue by doing the following:
   a. From the Kate window, select **File > Open**.
   b. Browse to and open the file **/etc/issue**.
   c. Scroll to the end of the file; then add the following text:

      ==========
      The DBA System
      ==========

   d. Add some space at the end of the file by pressing **Enter** twice.
   e. Save the file by selecting **File > Close**; then select **Save**.
2. Set an initial message of the day (upon a successful login) by editing the file /etc/motd:
   a. From the Kate window, select **File > Open**.
   b. Browse to and open the file **/etc/motd**.

      The file is empty.

   c. Enter the following message:

      Team,

      Welcome to the SLES 9 staging server.
      Your project files are in: /export/data2/db

      For questions call me at: 555-1212

      Thanks,

      Project Manager

   d. Add some space at the end of the file by pressing **Enter** twice.
   e. Delete any tabs or spaces at the beginning of the last line.
   f. Save the file by selecting **File > Close**; then select **Save**.

3. Close the Kate window by selecting **File > Quit**.

Part III: Create Users With YaST

Now that you've created a new group and changed the default home directory for new users, you are ready to create the database administrator user accounts.

Start by using YaST to create 2 user accounts.

Do the following:

1. From the KDE desktop, select the **YaST** icon; then log in as root by entering a password of **novell** and selecting **OK**.

   The YaST Control Center appears.

2. Select **Security and Users > Edit and create users**.

   The User and Group Administration dialog appears.

3. Create a new user account by selecting **Add**.

   The Add a New Local User dialog appears.

4. Enter the following information:
   - Full User Name: **Database Admin 1**
   - User Login: **dba1**
   - Password: **suse1**
   - Verify Password: **suse1**

5. Select **Password Settings**.

   A Password Settings for User dba1 dialog appears.

6. In the Expiration date field, enter a ***date*** (YYYY-MM-DD) for 5 days from today.

   For example, if today's date is 2005-07-01, you would enter **2005-07-06**.

7. When you finish, continue by selecting **Next**.

You are returned to the Add a New Local User dialog.

8. Select **Details**.

   An Add/Edit User Properties - Details dialog appears.

9. Make sure that the default Home Directory is **/export/home/dba1**.

   Because you changed the HOME= parameter in the file /etc/default to /export/home, YaST uses the setting to create a default directory for dba1 in /export/home/.

10. In the Additional Group Members list, make sure the group **dba** (at the top of the list) is selected.

11. When you finish, continue by selecting **Next**.

12. Select **Create**.

    You are returned to the User and Group Administration dialog.

13. Create a second database administrator account with the following parameters:

Table 2-1

Field	Value
Full User Name	**Database Admin 2**
User Login	**dba2**
Password	**suse2**
Groups	default groups plus the **dba** group
Expiration Date	5 days from today (such as 2005-07-06)
Home Directory	**/export/home/dba2**

14. When you finish creating both dba1 and dba2, complete the process by selecting **Finish**.

15. Verify that user account dba1 works:
    a. Switch to virtual console 2 by pressing **Ctrl+Alt+F2**.
    b. Log in by entering **dba1** and a password of **suse1**.

       Notice that the team welcome message (in the file /etc/motd) appears.
    c. Log out by entering **exit**.

       Notice that the message from the file **/etc/issue** appears.
    d. Log in again by entering **dba2** and a password of **suse2**; then log out by entering **exit**.
    e. Return to the KDE desktop by pressing **Ctrl+Alt+F7**.

    The user dba1 is the project lead for the database administrator group, and should have ownership rights to the directory /export/data2/db.
16. From the terminal window, make the user dba1 the owner of the directory /export/data2/db and dba the group owner by entering the following:

    chown dba1:dba /export/data2/db
17. Verify the changes by entering **ls -l /export/data2**.

Part IV: Create a User Account at the Command Line

Do the following:

1. Using the useradd command, create a user with the following information:

Table 2-2

Field	Value
Full User Name	**Database Admin 3**
User Login	**dba3**
Password	**suse3**

(continued) **Table 2-2**

Field	Value
Groups	default groups plus the **dba** group
Expiration Date	5 days from today (such as 2006-07-06)
Home Directory	**/export/home/dba3**

Enter the following (all on one line):

useradd -c "Database Admin 3" -d /export/home/dba3 -m -e *date* -G users,uucp,dialout,audio,video,dba dba3 ; passwd dba3

The parameter ***date*** should be the same expiration date you used for the other accounts (5 days from today).

This expiration date is for the user account; it is not the expiration date of the password. After the expiration date, the account becomes locked and cannot be accessed.

When the password expires, you are prompted to enter a new password.

After entering the command, you are prompted for a new password.

2. Enter a password of **suse3** twice.

   You are prompted that the password is too simple, but you are allowed to enter the password a second time.

   A password changed message appears.

3. Switch to virtual console 2 (**Ctrl+Alt+F2**) and login as **dba3**; then exit the login (**exit**) and return to the desktop (**Ctrl+Alt+F7**).

Part V: Manage Groups With YaST

Do the following:

1. From the YaST Control Center, select **Security and Users > Edit and create groups**.

   The User and Group Administration dialog appears.

2. Create a new group by selecting **Add**.

   An Add a New Local Group dialog appears.

3. In the Group Name field enter **dbabackup**; then select **Next**.

   You are returned to the User and Group Administration dialog.

4. Add all existing dba users to the group dbabackup:

   a. Select the group **dbabackup**; then select **Edit**.

      An Edit an Existing Local Group dialog appears.

   b. In the Members of this Group list, select **dba1**, **dba2**, and **dba3**.

   c. Continue by selecting **Next**.

   d. Complete the group configuration by selecting **Finish**.

   e. Switch to virtual console 2 (**Ctrl+Alt+F2**) and log in as **dba1** with a password of **suse1**.

   f. Verify that the new group was added by typing **groups**.

      Notice that dbabackup is listed for the user dba1.

   g. Log out by entering **exit**; then switch back to the desktop by pressing **Ctrl+Alt+F7**.

5. Make the group dbabackup the default group when creating a new user:

   a. From the YaST Control Center, select **Security and Users > Edit and create groups**.

   b. From the Expert Options drop-down list, select **Defaults for New Users**.

   c. From the Default Group drop-down list, select **dbabackup**.

You might need to scroll up to find the group in the list.

d. Continue by selecting **Next**.

e. Save the configuration change by selecting **Finish**.

6. Create a new user from **YaST** to test the configuration changes:

a. From the YaST Control Center, select **Security and Users > Edit and create users**.

b. Create a new user account by selecting **Add**.

c. Create a fourth database administrator account with the following parameters:

Table 2-3

Field	Value
Full User Name	**Database Admin 4**
User Login	**dba4**
Password	**suse4**
Groups	Default group (dbabackup) plus the **dba**, **audio**, **dialout**, **uucp**, and **video** groups
Expiration Date	5 days from today (such as 2005-07-06)
Home Directory	**/export/home/dba4**

Notice that the preselected default group for dba4 is dbabackup.

d. When you finish, save the new configuration by selecting **Finish**.

Part VI: Manage Groups From the Command Line

From the terminal window, do the following:

1. Create a group named party by entering **groupadd party**.

2. Add dba1, dba2, and dba3 to the group party by entering the following commands:

   usermod -G uucp,dialout,audio,video,dba,party,dbabackup dba1

   usermod -G uucp,dialout,audio,video,dba,party,dbabackup dba2

   usermod -G uucp,dialout,audio,video,dba,party,dbabackup dba3

   Make sure each command is on a single line with no spaces after the commas.

3. Verify that these users were added to the party group by entering the following commands:

   groups dba1
   groups dba2
   groups dba3

Part VII: Manage User Accounts From the Command Line

From the terminal window, do the following:

1. Check the entry in the file shadow for dba3 by entering the following:

   grep dba3 /etc/shadow

2. Lock the dba3 account using the command passwd:

   passwd -l dba3

3. Check the entry in shadow again by entering the following:

   grep dba3 /etc/shadow

   Notice the “!” after “dba3:”. This indicates that the account is locked.

4. Verify that the account is locked by switching to a virtual console and attempting to log in as dba3.

   A login incorrect message is displayed.

5. Return to the KDE desktop (**Ctrl+Alt+F7**) and unlock the dba3 account by entering the following:

   passwd -u dba3

6. Check the entry in shadow by entering the following:

   grep dba3 /etc/shadow

   Notice that the "!" has been removed from the account information.

7. Verify that the account is unlocked by switching to a virtual console and attempting to log in as dba3.

   You can now log in. Notice the number of failed attempts since the last login.

8. Log out as dba3 by entering **exit**.

9. Return to the KDE desktop (**Ctrl+Alt+F7**).

10. Change the information about the dba2 account in GECOS format by entering **passwd -f dba2**.

11. When you are prompted for a full name, continue by pressing **Enter**.

12. Enter the following information (as prompted):

    - Room Number: **Classroom**
    - Work Phone: **555-1212**
    - Home Phone: **444-1212**
    - Other: **Level 2 DBA**

13. Verify the information you added for dba2 by entering **finger dba2**.

14. Delete the group party by entering **groupdel party**.

15. Delete the account dba3 and the account home directory by entering **userdel -r dba3**.

    An informational message appears indicating that no crontab job exists for dba3.

16. View password policy information about the account dba2 by entering **chage -l dba2**.

Part VIII: Restore the Default Group Assignment For New Users

To restore the group users as the default group when creating a new user, do the following:

1. From the YaST Control Center, select **Security and Users > Edit and create groups**.
2. From the Expert Options drop-down list, select **Defaults for New Users**.
3. From the Default Group drop-down list, select **users**.
4. Continue by selecting **Next**.
5. Save the configuration changes by selecting **Finish**.
6. Close all open windows.

(End of Exercise)

Objective 3 Manage and Secure the Linux User Environment

Besides managing individual user accounts, you also need to know how to do the following to manage and secure the Linux user environment:

- Perform Administrative Tasks as root
- Delegate Administrative Tasks With sudo
- Set Defaults for New User Accounts
- Configure Security Settings

Perform Administrative Tasks as root

As a system administrator, you are advised to log in as a normal user and only switch to root to perform tasks that require root permissions.

To switch between a normal user and root while performing administrative tasks, you can do the following:

- Switch to Another User With su
- Switch to Another Group With newgrp
- Start Programs as Another User From KDE

Switch to Another User With su

You can use the command **su** (switch user) to assume the UID of root or of other users.

The following is the su syntax:

su [*options*] ...[-] [*user*[*argument*]]

For example, to change to the user geeko, enter **su geeko**; to change to the user root, enter **su root** or **su** (without a user name).

Root can change to any user ID without knowing the password of the user.

If you want to start a login shell when changing to the user root, you can enter **su -**.

To change to the user root and execute a single command, you can use the option **-c**:

su - -c "grep geeko /etc/shadow"

For additional information on the command su, enter **su --help**.

Switch to Another Group With newgrp

A user can be a member of many different groups, but only one GID is his *effective* (current) group at any one time. Normally this is the *primary group*, which is specified in the file /etc/passwd.

If a user creates directories or files, then they belong to the user and to the effective group.

You can change the effective group GID with the command newgrp or sg (such as **sg video**).

Only group members may perform this group change, unless a group password is defined. In this case, any user that knows the group password can make the change.

You can undo the change (return to the original effective GID) by entering **exit** or by pressing **Ctrl+D**.

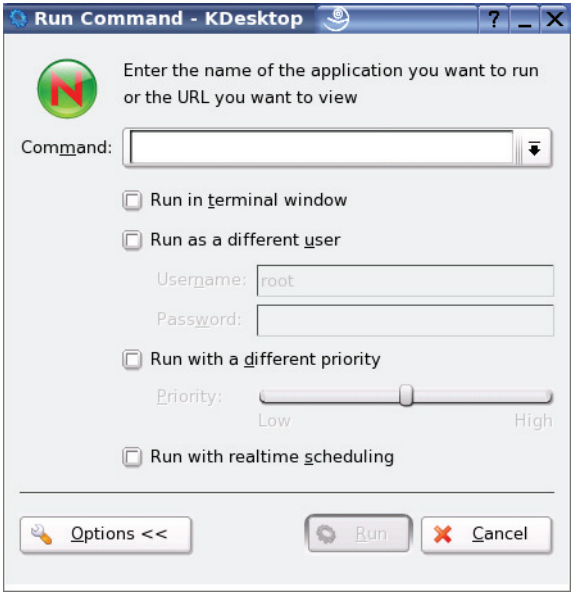
Start Programs as Another User From KDE

In KDE you can start any program with a different UID (as long as you know the password) by doing the following:

1. From the KDE desktop, open a command line dialog by pressing **Alt+F2**; then select **Options**.

   The following appears:

Figure 2-13



   From this dialog, you can enter a command that you want to run (or enter a URL to view).

   There are also several options, including the option to run the command as a different user.

2. Select **Run as a different user**: then enter the ***username*** (such as **root**) and the ***password***.

3. Enter the command you want to run as root; then select **Run**.

You can also enter **kdesu** ***program_name*** to start a program as root. You are prompted for the root password before the program starts.

Delegate Administrative Tasks With sudo

Sometimes it is necessary to allow a normal user access to a command which is usually reserved for root. For example, you might want a co-worker to take over tasks such as shutting down the computer and creating users while you are on vacation.

To enable a command to be run by a normal user, you can use the command sudo (as illustrated in the following):

```
geeko@earth:~ > sudo /sbin/shutdown -h now
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these two things:
#1) Respect the privacy of others.
#2) Think before you type.
Password:
```

You are prompted for a password, which is the user password.

As administrator, you can specify which commands a user can or cannot enter by configuring the file /etc/sudoers. You can modify the configuration by using the command visudo.

In SLES 8, sudo expected the password of the user executing sudo. To change to this default setting, put comment signs (#) in front of the lines “Defaults targetpw” and “%users ALL=(ALL) ALL” in the file /etc/sudoers using the command visudo.

The following is the general syntax of an entry in the configuration file:

user/group host = command1 , command2 ...

For example

```
geeko ALL = /sbin/shutdown
```

In this example, the user geeko is able to carry out the command **/sbin/shutdown** with the permissions of root on all computers (**ALL**).

The following is a more complex example that illustrates the flexibility of sudo:

Figure 2-14

```
  User_Alias  ADMINS     = tux, geeko
2 User_Alias  WEBMASTER  = john
  User_Alias  SUBSTITUTE = olli, klaas
4
  # Cmnd alias specification
6
  Cmnd_Alias  PRINTING = /usr/sbin/lpc, /usr/bin/lprm
8 Cmnd_Alias  SHUTDOWN = /sbin/shutdown
  Cmnd_Alias  APACHE   = /etc/init.d/apache
10

12 # User privilege specification
   root    ALL=(ALL) ALL
14
   ADMINS     ALL = NOPASSWD: !/usr/bin/passwd, /usr/bin/passwd [A-z]*,
16 !/usr/bin/passwd root
   WEBMASTER  ALL = APACHE
18 SUBSTITUTE ALL = SHUTDOWN, PRINTING
```

Lines 1 to 9 define aliases. You can do this for the following:

- Users with User_Alias (lines 1–3)
- Commands with Cmnd_Alias (lines 7–9)
- Hosts with Host_Alias

Lines 14–17 in this example show how these aliases can be used in the actual rules:

- **ADMINS.** This is the User_Alias for the users tux and geeko (see line 1).

The following are additional parameters:

- **!/usr/bin/passwd, /usr/bin/passwd [A-z]*.** This indicates that both users are allowed to run the command passwd with one single argument and change the passwords for user accounts.
- **!/usr/bin/passwd root.** This indicates that both users are not allowed to change the password for root. However, they can change the passwords of other users.

With this configuration, tux and geeko could still lock out root by entering **sudo /usr/bin/passwd root -l**.

- **WEBMASTER.** This is the User_Alias for the user account john (see line 2). This user can start and stop the web server (APACHE).
- **SUBSTITUTE.** This is the User_Alias for the user accounts olli and klaas (see line 3). These users can execute commands summarized in sections SHUTDOWN and PRINTING (see lines 7 and 8).

For additional documentation and configuration examples, enter **man 5 sudoers**.

Set Defaults for New User Accounts

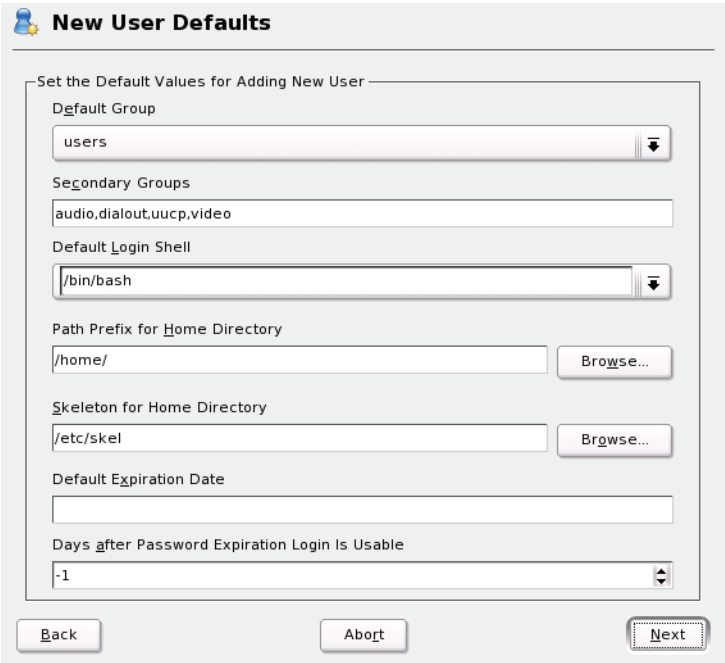
You can use YaST to select default settings to be applied to new user accounts by doing the following:

1. From the KDE desktop, start the YaST Edit and create users module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Security and Users > Edit and create users**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 users**.
2. Select **Expert Options > Defaults for New Users**.

   The following appears:

Figure 2-15



3. Enter or edit information in the following fields:
   - **Default Group.** From the drop-down list select the primary (default) group.
   - **Secondary Groups.** Enter a list of secondary groups (separated by commas) to assign to the user.
   - **Default Login shell.** From the drop-down list select the default login shell (command interpreter) from the shells installed on your system, or enter your own path to the shell.
   - **Default Home.** Enter or browse to the initial path prefix for a new user's home directory. The user's name will be appended to the end of this value to create the default name of the user's home directory.
   - **Skeleton Directory.** Enter or browse to the skeleton directory. The contents of this directory will be copied to the user's home directory when you add a new user.
   - **Default Expiration Date.** Enter the date on which the user account is disabled. The date must be in the format YYYY-MM-DD.

     Leave the field empty if this account never expires.
   - **Days after Password Expiration Login Is Usable.** This setting enables users to log in after passwords expire. Set how many days after a password expires that login is allowed.

     Enter **-1** for unlimited access.
4. Save the configuration settings by selecting **Next > Finish**.

Configure Security Settings

Yast provides a Security Settings module that lets you configure the following local security settings for your SUSE Linux Enterprise Server:

- Password settings
- Boot configuration
- Login settings
- User creation settings
- File permissions

You can select from (or modify) three preset levels of security, or create your own customized security settings to meet the requirements of your enterprise security policies and procedures.

To use the Security Settings module, do the following:

1. From the KDE desktop, start the YaST Security Settings module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Security and Users > Security Settings**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 security**.

The following appears:

Figure 2-16

From this dialog, you can select one of the following preset configurations:

- **Level 1 (Home Workstation).** Select for a home computer not connected to any type of a network.
- **Level 2 (Networked Workstation).** Select for a computer connected to any type of a network or the Internet.
- **Level 3 (Network Server).** Select for a computer that provides any type of service (network or otherwise).

You can also select **Details** or **Custom Settings** to modify an existing security level or create your own configuration.

2. Do one of the following:
   - Select a preconfigured ***security level*** setting; then configure your server by selecting **Finish**.

     or
   - Select a preconfigured **security level** setting; then customize the level by selecting **Details**.

     or
   - Create your own customized security level by selecting **Custom Settings**; then select **Next**.

The remainder of these steps guide you through customizing a preconfigured security level or creating a customized security level.

The following appears.

Figure 2-17

From this dialog, you can select or enter the following password settings (mainly stored in /etc/login.defs):

- **Checking New Passwords.** It is important to choose a password that cannot be found in a dictionary and is not a name or other simple, common word. By selecting this option, you enforce password checking in regard to these rules.
- **Plausibility Test for Passwords.** Passwords should be constructed using a mixture of characters. This makes it very difficult to guess the password. Select this option to enable additional checks.
- **Password Encryption Method.** From the drop-down list, select one of the following encryption methods:
  - **DES.** This is the Linux default method. It works in all network environments, but it restricts you to passwords no longer than eight characters. If you need compatibility with other systems, select this method.
  - **MD5.** This encryption method allows longer passwords and is supported by all current Linux distributions, but not by other systems or older software.
  - **Blowfish.** This encryption method is similar to MD5, but uses a different algorithm to encrypt passwords. It is not yet supported by many systems. A lot of CPU power is needed to calculate the hash, which makes it difficult to crack passwords with the help of a dictionary.
- **Number of Significant Characters in the Password.** You can only set this number for DES encryption. The default is **8** characters.

  This option is ignored for the other encryption methods (MD5=**127**; Blowfish=**72**).

- **Minimum Acceptable Password Length.** Enter the minimum number of characters for an acceptable password. If a user enters fewer characters, the password is rejected.

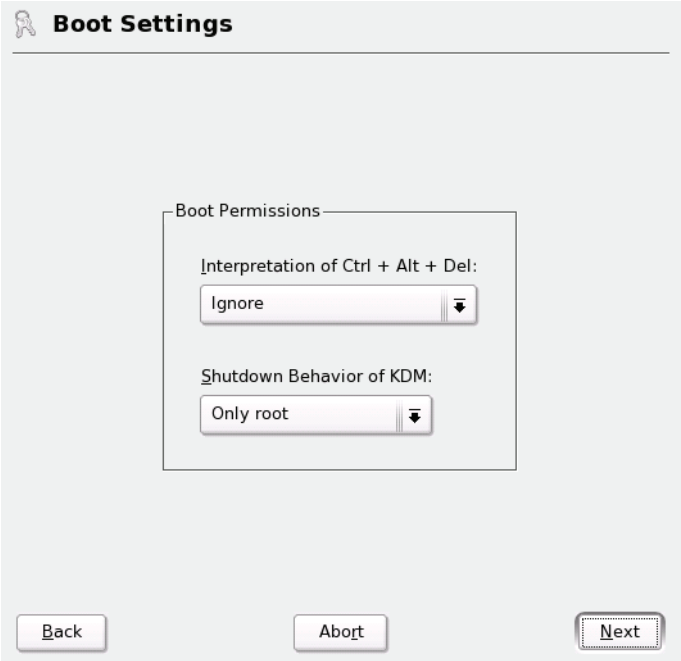
  Entering **0** disables this check.
- **Days to Password Change Warning.** Enter the minimum and maximum times for warning users that the password must be changed.
- **Days to Password Change Warning.** Minimum refers to the number of days that have to elapse before a password can be changed again. Maximum is the number of days after which a password expires and must be changed.

Although root receives a warning when setting a password, she can still enter a bad password despite the above settings.

3. When you finish configuring password settings, continue by selecting **Next**.

The following appears:

Figure 2-18



From this dialog, you can select the following boot settings (which update the file /etc/inittab):

- **Interpretation of Ctrl + Alt + Del.** When someone at the console presses the **Ctrl+Alt+Del** keystroke combination, the system usually reboots.

  Sometimes you want to have the system ignore this keystroke combination, especially when the system serves as both workstation and server.

  You can select from **Ignore**, **Reboot**, or **Halt**. If you select Halt, the system shuts down.

- **Shutdown Behavior of KDM.** You use this option to set who is allowed to shut down the computer from KDM.

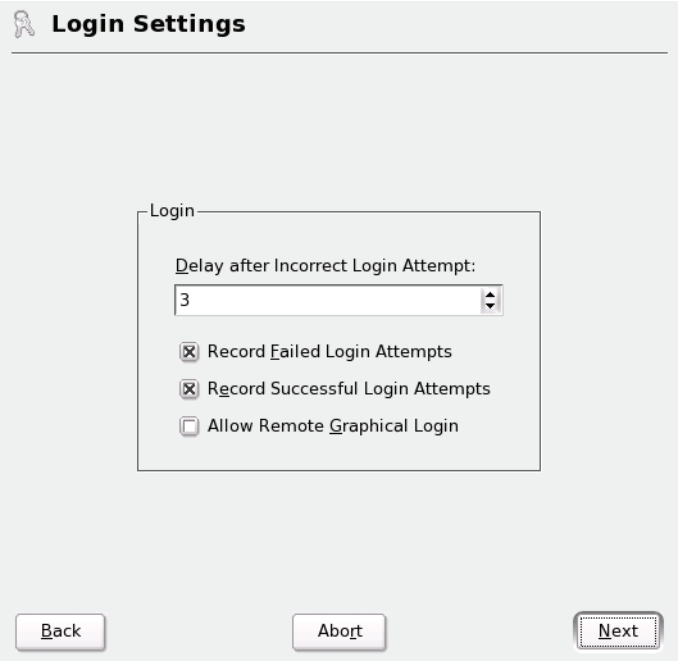
You can select from **Only root**, **All users**, **Nobody**, **Local users**, and **Automatic**.

If you select Nobody, you can only shut down the system from a text console.

4. When you finish configuring boot settings, continue by selecting **Next**.

The following appears:

Figure 2-19



From this dialog, you can enter and select the following login settings (mainly stored in /etc/login.defs):

- **Delay after Incorrect Login Attempt.** Following a failed login attempt, there is typically a waiting period of a few seconds before another login is possible. This makes it more difficult for password crackers to log in.

This option lets you adjust the time delay before another login attempt.

Make the time small enough so users do not need to wait too long to retry if a password is mistyped. A reasonable value is **3** seconds.

- **Record Failed Login Attempts.** It is useful for you to know if somebody tried to log in and failed, especially when that person is trying to guess other users' passwords.

  Select this option to specify whether failed login attempts should be recorded in /var/log/faillog. To view failed login attempts, enter the command **faillog**.

- **Record Successful Login Attempts.** Logging successful login attempts can be useful, especially in warning you of unauthorized access to the system (such as a user logging in from a different location than normal).

  Select this option to record successful login attempts in the file /var/log/wtmp. You can use the command last to view who logged in at what time.

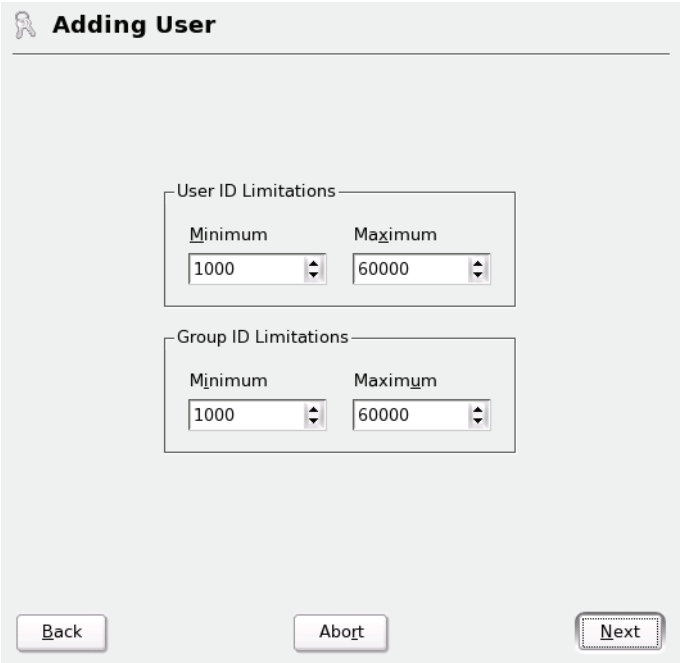
- **Allow Remote Graphical Login.** You can select this option to allow other users access to your graphical login screen via the network.

  Because this type of access represents a potential security risk, it is inactive by default.

5. When you finish configuring login settings, continue by selecting **Next**.

The following appears:

Figure 2-20



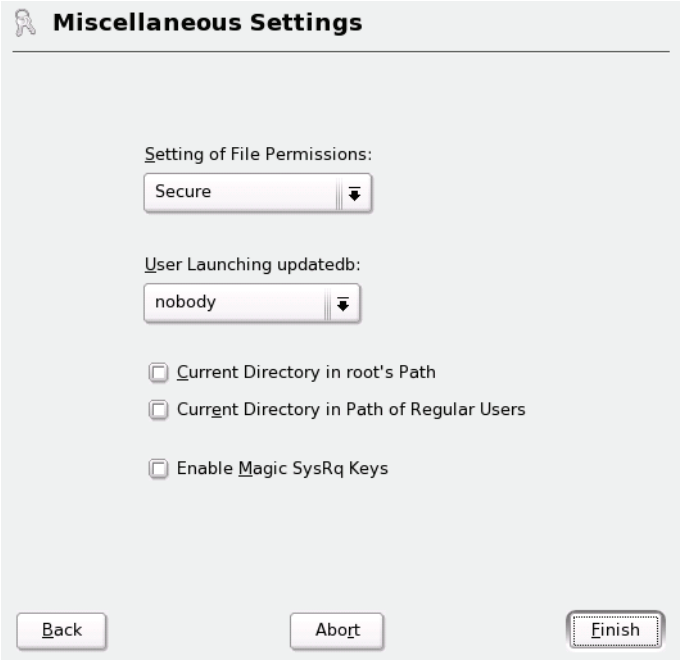
From this dialog, you can enter the following ID settings:

- **User ID Limitations.** Enter a minimum and maximum value to configure a range of possible user ID numbers. A minimum of 500 is suitable for users.
- **Group ID Limitations.** Enter a minimum and maximum value to configure a range of possible group ID numbers.

6. When you finish configuring user and group ID limitations, continue by selecting **Next**.

The following appears:

Figure 2-21



From this dialog, you can select the following miscellaneous global settings:

- **Setting of File Permissions.** Settings for the permissions of certain system files are configured in **/etc/permissions.secure** or **/etc/permissions.easy**.

  From the drop-down list, select one of the following:

  - **Easy.** Select this option to allow read access to most of the system files by users other than root.

- **Secure.** Select this option to make sure that certain configuration files (such as /etc/ssh/sshd_config) can only be viewed by the user root. Some programs can only be launched by root or by daemons, not by an ordinary user.
- **Paranoid.** Select this option for an extremely secure system. All SUID/SGID-Bits on programs have been cleared. Remember that some programs might not work or not work correctly, because users no longer have the permissions to access certain files.

Running SuSEconfig sets these permissions according to the settings in the /etc/permissions.* files. This fixes files with incorrect permissions, whether this occurred accidentally or by intruders.

- **User Launching updatedb.** If the program updatedb is installed, it automatically runs on a daily basis or after booting. It generates a database (**locatedb**) in which the location of each file on your computer is stored.

  You can search this database with the utility locate (enter **man locate** for details).

  From the drop-down list, select one of the following:

  - **nobody.** Any user can find only the paths in the database that can be seen by any other (unprivileged) user.
  - **root.** All files in the system are added into the database.

- **Current Directory in root's Path** and **Current Directory in the Path of Regular Users.** On a DOS system, DOS first searches for executable files (programs) in the current directory, and then in the current path variable. This is not the case on a Linux system.

  Some systems set up a “workaround” by adding the dot (“.”) to the search path, which enables files in the current path to be found and executed.

If you deselect these options, users must always launch programs in the current directory by adding "./" (such as **./configure**).

If you select these options, the dot (".") is appended to the end of the search path for root and users, allowing them to enter a command in the current directory without appending "./".

Selecting these options can be very dangerous because users can accidentally launch unknown programs in the current directory instead of the usual system-wide files.

In addition, selecting these options allows execution of Trojan Horses, which can exploit this weakness and intrude your system.

- **Enable Magic SysRq Keys.** Selecting this option gives you some control over the system even if it crashes (such as during kernel debugging). For details, see /usr/src/linux/Documentation/sysrq.txt.

7. When you finish configuring the miscellaneous settings, save the settings and run SuSEconfig by select **Finish**.

Exercise 2-3 Configure the Password Security Settings

Do the following:

1. Open a terminal window.
2. Check the trap setting for the Ctrl+Alt+Del keystroke in the file /etc/inittab by entering **grep ctrlaltdel /etc/inittab.**

   Notice that the trap is set by default to “shutdown -r” for restart.
3. From the KDE desktop, select the **YaST** icon; then enter a password of **novell** and select **OK**.
4. Select **Security and Users > Security settings**.

   The Local Security Configuration dialog appears.
5. Make sure **Custom Settings** is selected; then select **Next**.

   The Password Settings dialog appears.
6. From the Password Encryption Method drop-down list, select **MD5**.
7. Continue by selecting **Next**.

   The Boot Settings dialog appears.
8. From the Interpretation of Ctrl + Alt + Del drop-down list, select **Halt**.
9. Continue by selecting **Next**.

   The Login Settings dialog appears.
10. Accept the default settings by selecting **Next**.

    The Adding User dialog appears.
11. Accept the default settings by selecting **Next**.

    The Miscellaneous Settings dialog appears.
12. Deselect the following:
    - **Current Directory in root’s Path**
    - **Current Directory in Path of Regular Users**

13. Configure the system for the new settings by selecting **Finish**.

    To test the change, you must first activate the new configuration by rebooting the system or by entering (as root) **init -q** (reload the /etc/inittab file) in a terminal window.

14. From the terminal window, su to **root** (**su -**) with a password of **novell**.

15. Reboot the system by entering **init 6**.

16. When the system reboots, log in to the KDE desktop as **geeko** with a password of **N0v3ll**.

17. From a terminal window, verify that the Ctrl+Alt+Del setting has changed by entering **grep ctrlaltdel /etc/inittab**.

    Notice that the trap is now set to “shutdown -h” for halt instead of “shutdown -r” for restart.

18. Test this setting by pressing **Ctrl+Alt+F2** and log in as **root**; then press **Ctrl+Alt+Del**.

    The system shuts down instead of restarting.

19. Turn on your computer and log in to the KDE desktop as **geeko**.

20. (Optional) Use the YaST Security settings module to change the default for **Ctrl+Alt+Del** back to restart.

(End of Exercise)

Objective 4 Secure Files and Directories With Permissions

Although users might have access to navigate through the files and directories on a Linux system, you can limit what they can do with those files and directories by using access permissions.

To set permissions for files and directories, you need to know the following:

- Permissions and Permission Values
- How to Set Permissions From the Command Line
- How to Set Permissions From a GUI Interface
- How to Modify Default Access Permissions
- How to Configure Special File Permissions
- How to Configure Additional File Attributes for ext2

Permissions and Permission Values

You can assign the following 3 permissions to a file or directory:

- **Read (r).** This permission allows the file to be read or the contents of a directory to be listed.
- **Write (w).** This permission allows a file to be modified. It allows files to be created or deleted within a directory.
- **Execute (x).** This permission allows a file to be executed. It allows access to a directory.

You can use the command **ls -l** to display the contents of the current directory with the assigned permissions for each file or subdirectory.

For example, entering **ls -l** displays the following permissions for myfile.txt:

Figure 2-22

Owner Group Others

```
[root@RCE tmp]# ls -l
total 40
-rw-r--r--    1 root     root        12304 Dec 12 03:44 install.log
-rw-r--r--    1 root     root            0 Dec 12 03:17 install.log.syslog
-rwxr-xr--    1 root     root            6 Dec 12 14:55 myfile.txt
drwx------    2 root     root         4096 Dec 12 03:32 orbit-root
-rw-rw-rw-    1 apache   apache          6 Dec 12 11:48 rcq-runner.pid
-rw-------    1 apache   apache       6255 Dec 12 11:53 sess_51e5c8d48f41b4a3d0e
ef964966f0775
-rw-------    1 root     root            0 Dec 12 11:48 session_mm_apache0.sem
-rw-------    1 root     root            0 Dec 12 11:48 session_mm_cgi0.sem
drwxr-xr-x    8 201      201          4096 Dec 12 10:51 vmware-tools-distrib
[root@RCE tmp]# _
```

You can also view permissions from a file manager or browser tool.

For example, you can use the Detailed List View in Konqueror to view permissions, owner, and group for each directory or file:

Figure 2-23

Name	Size	File Type	Modified	Permissions	Owner	Group	Link
bin	48 B	Folder	2004-07-09 08:22	rwxr-xr-x	geeko	users	
Desktop	376 B	Folder	2004-07-24 08:31	rwx------	geeko	users	
Documents	80 B	Folder	2004-07-09 08:22	rwxr-xr-x	geeko	users	
public_html	80 B	Folder	2004-07-09 08:22	rwxr-xr-x	geeko	users	
snapshot1.png	51.7 KB	PNG Image	2004-07-26 13:19	rw-r--r--	geeko	users	
snapshot2.png	14.5 KB	PNG Image	2004-07-24 14:44	rw-r--r--	geeko	users	

How to Set Permissions From the Command Line

You can modify a file or directory's permissions and ownership from the command line by using the following:

- chmod
- chown and chgrp

chmod

You can use this command to add, remove, or assign permissions assigned to a file or directory. Both the owner of a file and root can use this command.

The following are examples of using the command chmod:

Table 2-4

chmod command	Description
chmod u+x	The owner is given permission to execute the file. The permissions r and w stay as they are.
chmod g=rw	All group members can read and write. If the members had the execute permission before, it is removed.
chmod u=rwx	The owner receives all permissions.
chmod u=rwx,g=rw,o=r	The owner has all permissions, the group has read and write permissions, and all others have read permission.
chmod +x	All users (owner, group, and others) receive executable permission, depending on umask.

For example, entering the following chmod command lets all users in the group users write to the file hello.txt:

```
geeko@earth:~ > ls -la hello.txt
-rw-r--r-- 1 geeko users 0 2004-04-06 12:40 hello.txt
geeko@earth:~ > chmod g+w hello.txt
geeko@earth:~ > ls -la hello.txt
-rw-rw-r-- 1 geeko users 0 2004-04-06 12:40 hello.txt
```

With the option -R (recursive) and a specified directory, you can change the access permissions of all files and subdirectories under the directory.

Besides using the letters rwx to indicate permissions, you can also use groups of numbers.

Every file and directory in a Linux system has a numerical permission value assigned to it. This value has 3 digits.

The first digit represents the permissions assigned to the file or directory owner. The second digit represents the permissions assigned to the group associated with the file or directory. The third digit represents the permissions assigned to others.

Each digit is the sum of the following 3 values assigned to it:

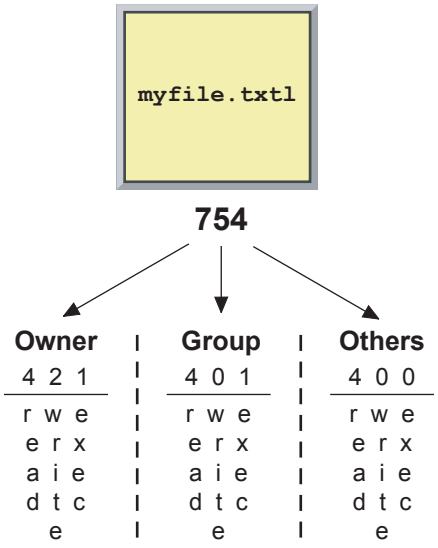
- Read: **4**
- Write: **2**
- Execute: **1**

For example, suppose a file named myfile.txt has **754** permissions assigned to it.

This means the owner of the file has read, write, and execute permissions (**4+2+1**), the group associated with the file has read and execute permissions (**4+1**), and others have read permissions (**4**).

This is illustrated in the following:

Figure 2-24



The following are examples of using numerical values with chmod:

Table 2-5

chmod command	Description
chmod 754 hello.txt	The owner has all permissions, the group has read and execute permissions, and all others have the read permission.
chmod 777 hello.txt	All users (user, group, and others) receive all permissions.

chown and chgrp

These commands change the owner or group assigned to a file or directory.

As user root, you can use the command chown to change the user and group affiliation of a file using the following syntax:

chown *new_user.new_group file*

For example, in the following, root changes ownership of the file hello.txt from geeko to the user newbie:

```
earth:/tmp # ls -la hello.txt
-rw-r--r-- 1 geeko users 0 2004-04-06 12:43 hello.txt
earth:/tmp # chown newbie.users hello.txt
earth:/tmp # ls -la hello.txt
-rw-r--r-- 1 newbie users 0 2004-04-06 12:43 hello.txt
earth:/tmp #
```

To change only the owner (and not the group), use the following:

chown *new_user file*

To change only the group (and not the user), use the following:

chown *.new_group file*

For example, the following command is used to limit access to the file list.txt to members of the group advanced:

```
earth:/tmp # ls -la list.txt
-rw-r----- 1 geeko users 0 2004-04-06 12:43 list.txt
earth:/tmp # chown .advanced list.txt
earth:/tmp # ls -la list.txt
-rw-r----- 1 geeko advanced 0 2004-04-06 12:43 list.txt
earth:/tmp #
```

Although the group has changed, the owner permissions remain the same (the owner and root can still access the file).

As user root, you can also change the group affiliation of a file with the command chgrp using the following syntax:

chgrp *.new_group file*

A normal user can only use the command chown to change his or her file to a new group, as in the following:

chown *.new_group file*

Of course, the user can also do the same with chgrp, as in the following:

chgrp *new_group file*

Users can change the group affiliation of their files only if they are a member of the new group.

How to Set Permissions From a GUI Interface

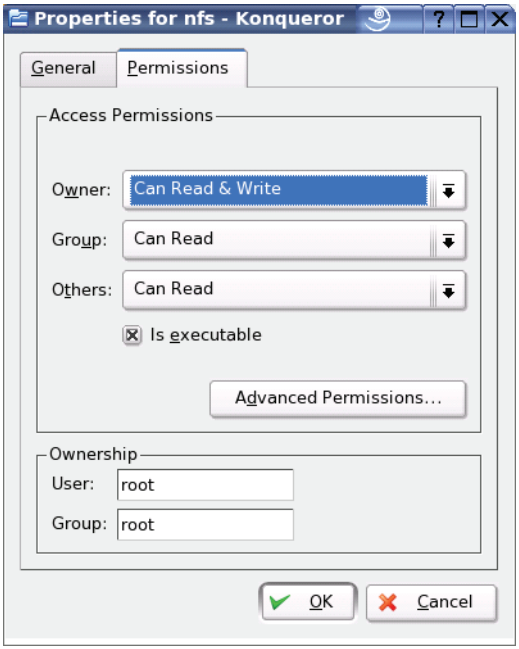
Besides using commands (such as chmod or chown), you can modify a file or directory's permissions from a GUI interface (normally a file browser).

For example, you can use Konqueror in KDE to change permissions by doing the following:

1. Start Konqueror; then browse to the ***file*** or ***directory*** (do not open it).
2. Right-click the ***file*** or ***directory*** you want to modify; then select **Properties**.
3. Select the **Permissions** tab.

A dialog similar to the following appears:

Figure 2-25



From this dialog, you can change the Read (r) and Write (w) permissions for Owner, Group, and Others by selecting the appropriate option (**Can Read** or **Can Read & Write**) from the drop-down lists.

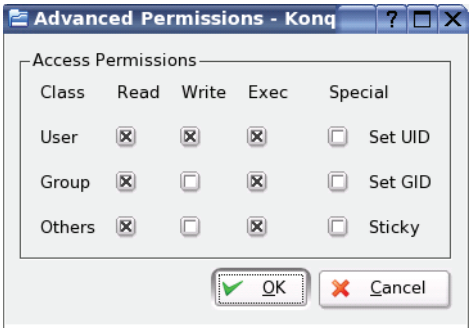
You can deny all permissions (equivalent to **0**) by selecting **Forbidden**.

You can also modify the user and group ownership of the file or directory by entering a user or group in the appropriate field.

4. Modify the permissions and ownership as desired.
5. (Optional) Modify individual permissions by doing the following:
   a. Select **Advanced Permissions**.

The following appears:

Figure 2-26



a. Select the ***permissions*** you want to set, and then finish by selecting **OK**.

6. When you finish configuring permissions for the file or directory, save the configuration by selecting **OK**.

How to Modify Default Access Permissions

If the default settings are not changed, files are created with the access mode **666** and directories with **777** by default.

To modify (restrict) these default access mode settings, you can use the command *umask*. You use this command with a 3-digit numerical value such as **022**.

How can you calculate the default setting for file and directory permissions from the umask value? The permissions set in the umask are removed from the default permissions.

For example, entering **umask 022** ***directory_name*** or **umask 022** ***filename*** gives the following result:

Table 2-6

	Directories			Files		
Default Permissions	rwx	rwx	rwx	rw-	rw-	rw-
	7	7	7	6	6	6
umask	---	-w-	-w-	---	-w-	-w-
	0	2	2	0	2	2
Result	rwx	r-x	r-x	rw-	r--	r--
	7	5	5	6	4	4

Entering **umask 023** ***directory_name*** or **umask 023** ***filename*** gives the following result:

Table 2-7

	Directories			Files		
Default Permissions	rwx	rwx	rwx	rw-	rw-	rw-
	7	7	7	6	6	6
umask	---	-w-	-wx	---	-w-	-wx
	0	2	3	0	2	3
Result	rwx	r-x	r--	rw-	r--	r--
	7	5	4	6	4	4

In the second example (umask 023), the **x** permission in the umask does not have any effect on the file permissions, as the x permission is missing in the default setting (rw- rw- rw-, 666).

By entering **umask 077** you restrict access to the owner and root only; the group and others do not have any access permissions.

To make the umask setting permanent, you can change the value of umask in the system-wide configuration file /etc/profile.

If you want the setting to be user-specific, enter the value of umask in the file .bashrc in the home directory of the respective user.

How to Configure Special File Permissions

The following 3 attributes are used for special circumstances (use the uppercase letter in the absence of the execute bit):

Table 2-8

Letter	Number	Name	Files	Directories
t or T	1	Sticky bit	not applicable	Users can only delete files when they are the owner, or when they are root or owner of the directory. This is usually applied to the directory /tmp/.
s or S	2	SGID (set GroupID)	When a program is run, this sets the group ID of the process to that of the group of the file.	Files created in this directory belong to the group to which the directory belongs and not to the primary group of the user. New directories created in this directory inherit the SGID bit.

(continued) **Table 2-8**

Letter	Number	Name	Files	Directories
s or S	4	SUID (set UserID)	Sets the user ID of the process to that of the owner of the file when the program is run.	Not applicable

You set the sticky bit with chmod, either via the permissions of others (such as **chmod o+t /tmp**) or numerically (such as **chmod 1777 /tmp**).

The sticky bit on older Unix systems enabled the storing of an executable program in memory after it had been terminated, so it could be quickly restarted. However, with modern Unix and Linux systems, this only affects directories.

The sticky bit is listed in the permissions for Others (t), as in the following:

```
geeko@earth:~ > ls -ld /tmp
drwxrwxrwt 15 root root 608 2004-04-06 12:45 /tmp
geeko@earth:~ >
```

The following is an example for SUID:

```
geeko@earth:~ > ls -l /usr/bin/passwd
-rwsr-xr-x 1 root shadow 79765 2004-03-24 12:19 /usr/bin/passwd
geeko@earth:~ >
```

The following is an example for SGID:

```
geeko@earth:~ > ls -l /usr/bin/wall
-rwxr-sr-x 1 root tty 10192 2004-03-22 05:24 /usr/bin/wall
geeko@earth:~ >
```

If the attributes SUID or SGID are set, the programs are carried out with the privileges the owner (in the example for SUID above: root) or the group (in the example for SGID above: tty) have.

If root is the owner of the program, the program is carried out with the permissions of root. Unfortunately, there is a certain security risk in doing this.

For example, it could be possible for a user to take advantage of an error in the program, retaining root privileges after the process has been ended.

How to Configure Additional File Attributes for ext2

The file attributes introduced so far do not cover all the possible requirements of an operating system. For that reason, the following additional file permissions have been included in the ext2 file system (and are available in ext3):

Table 2-9

Attribute	Description
A	(atime) Controls whether an atime entry for this file is processed on the hard drive. For example, If it is set, hard drive activities for laptop computers can be reduced.
a	(append) Data can only be appended to this file. The file cannot be overwritten or deleted.
d	(no dump) If this attribute is set, the file is not saved when a backup is made with dump.
i	(immutable) The file can no longer be processed or deleted and no hard link can be set to it.
S	(synchronous update) Causes a synchronous writing to the hard drive (without saving in the buffer). This can slow down the machine, because all caching processes for this file are switched off.

You can set all these attributes as root with the command chattr (change file attributes) and display them with the command lsattr (list file attributes).

You use the following syntax to set the attributes:

Table 2-10

Syntax	Description
+	Adds attributes to the existing ones
-	Removes attributes
=	Sets only these attributes for one file

Options for chattr and lsattr include the following:

Table 2-11

Option	Description
-R	**chattr.** Runs command in subdirectories as well **lsattr.** Lists directories and their contents
-v	**chattr.** Changes the stored version number of the file **lsattr.** Lists the version number of the file
-a	**lsattr.** Lists all files in the directory (including hidden ones)

Exercise 2-4 Set Permissions for Files and Directories From the Command Line

Do the following:

1. Open a terminal window, and su to root (**su -**) with a password of **novell**.
2. Create the directory /files/ by entering **mkdir /files**.
3. Change to the directory /files/ by entering **cd /files**.
4. Use the mkdir command to create the subdirectories **private** and **public** under /files/.
5. Change the permissions on the private directory so that only root has read, write, and execute permissions by entering the following:

   chmod 700 private
6. Change permissions on the public directory so that everyone has rights to the directory by entering the following:

   chmod 777 public
7. Verify the changes by entering **ls -l**.
8. Switch to virtual terminal 3 by pressing **Ctrl+Alt+F3**.
9. Login as **geeko** with a **N0v3ll** password.
10. Switch to the directory /files/ by entering **cd /files**.
11. Try to create a file named geeko in the private directory by entering **touch private/geeko**.

    Notice that permission is denied.
12. Try to create a file named geeko in the public directory by entering **touch public/geeko**.
13. Verify that the file is created by entering **ls public**.
14. Change to the public directory by entering **cd public**.

15. List the permissions of the file geeko by entering **ls -l geeko**.

    Notice that the group users and other have only read permission for the file.

16. Change permissions so that the users group has write permissions and other does not have any permissions by entering the following:

    chmod g+w,o-r geeko

17. Verify the change by entering **ls -l**.

18. Log out as geeko by entering **exit** (or by pressing **Ctrl+D**); then return to the KDE Desktop (**Ctrl+Alt+F7**).

19. Close the terminal window.

(End of Exercise)

Objective 5 Configure User Authentication With PAM

Linux uses PAM (Pluggable Authentication Modules) in the authentication process as a layer that communicates between users and applications.

By providing system-wide access to applications through its authentication modules, PAM lets you configure and change authentication methods between users and individual applications from centrally managed modules.

Whenever a new authentication method is needed (such as a fingerprint scan instead of a username/password) for an application, you simply reconfigure or create a PAM module for use by the application.

To understand how to configure PAM, you need to know the following:

- Location and Purpose of PAM Configuration Files
- PAM Configuration File Structure
- PAM Configuration File Examples
- PAM Documentation Resources

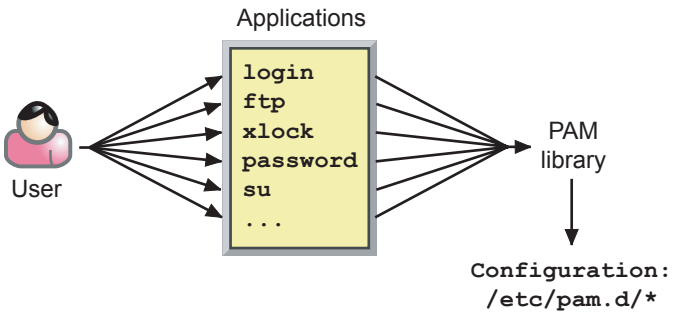
Location and Purpose of PAM Configuration Files

PAM provides a variety of modules—each one with a different purpose. For example, one module checks the password, another verifies the location from which the system is accessed, and another reads user-specific settings.

Every program that relies on the PAM modules has its own configuration file in the directory **/etc/pam.d/*program_name***.

These files define the PAM modules that are use for authentication, and include the following:

Figure 2-27



For example, the configuration file for the program passwd is called /etc/pam.d/passwd.

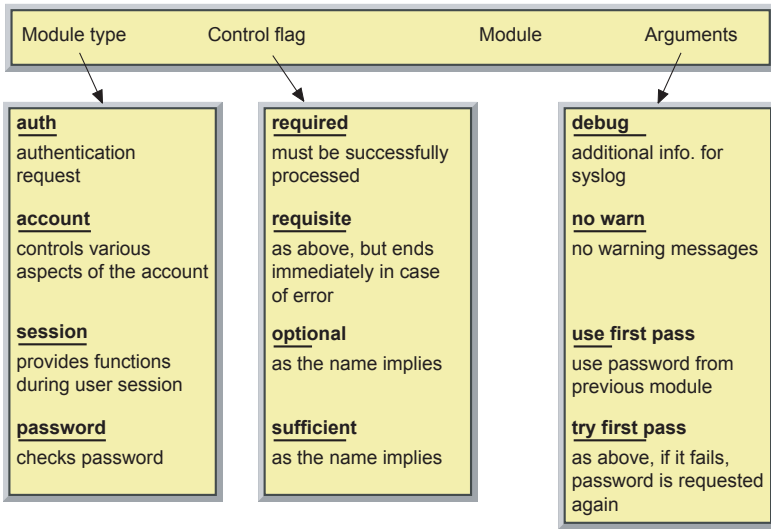
In addition, there are global configuration files for most PAM modules in /etc/security/, which define the exact behavior of these modules. These include modules such as pam_env.conf, pam_pwcheck.conf, pam_unix2.conf, and time.conf.

Every application that uses a PAM module actually calls a set of PAM functions, which then process the information in the various configuration files and return the result to the calling application.

PAM Configuration File Structure

Each line in a PAM configuration file contains a maximum of 4 columns:

Figure 2-28



The following describes the purpose of each column:

- **Module type.** PAM recognizes 4 types of modules:
  - **auth.** This module type checks the user's authenticity. This is traditionally done by querying a password, but it can also be achieved with the help of a chip card or through biometrics (fingerprints or iris scan).
  - **account.** This module type verifies whether the user has general permission to use the requested service. For example, this type of check should be performed to ensure that no one can log in under the user name of an expired account.

- **session.** This module type is responsible for managing and configuring user sessions. Sessions are started before and after authentication to register login attempts in system logs and to configure the user's specific environment (such as mail accounts, home directory, and system limits).
- **password.** This module type is used to enable the change of an authentication token. In most cases, this is a password.

■ **Control flag.** This column contains control flags that influence the behavior of PAM modules:

- **required.** A module with this flag must be successfully processed before the authentication can proceed.

  After the failure of a module with the required flag, all other modules with the same flag are processed before the user receives a message about the failure of the authentication attempt.
- **requisite.** A module with this flag must also be processed successfully. In case of success, other modules are subsequently processed, just like modules with the required flag.

  However, in case of failure the module gives immediate feedback to the user and no further modules are processed.

  You can use the requisite flag as a basic filter, checking for the existence of certain conditions that are essential for a correct authentication.
- **optional.** The failure or success of a module with this flag does not have any direct consequences.

  You can use this flag for modules that are only intended to display a message (such as telling a user that mail has arrived) without taking any further action.

- **sufficient.** After a module with this flag has been successfully processed, the calling application receives an immediate message about the success and no further modules are processed (provided there was no preceding failure of a module with the required flag).

  The failure of a module with the sufficient flag has no direct consequences. In other words, any subsequent modules are processed in their respective order.

- **Module.** You do not need to include the module, as long as the module is located in the default directory /lib/security/.

For all 64 bit platforms supported by SUSE Linux, the default directory is /lib64/security/.

- **Arguments (options).** You can include options in this column for the module, such as **debug** (enables debugging) or **nullok** (allows the use of empty passwords).

PAM Configuration File Examples

The following are examples of PAM configuration files for the modules pam_securetty.so and pam_nologin.so that illustrate what you can do to provide secure access to applications:

- **pam_securetty.so.** You can configure this module to determine which terminals can be regarded as secure. The user root can log in only at these terminals.

  The following line in the file /etc/pam.d/login activates the module pam_securetty.so for the program login:

  ```
  auth required pam_securetty.so
  ```

In the file /etc/securetty you specify the secure terminals, as in the following:

```
earth:~ # cat /etc/securetty
#
# This file contains the device names of tty lines (one per line,
# without leading /dev/) on which root is allowed to login.
#
tty1
tty2
tty3
tty4
tty5
tty6
...
```

When this module is active, you cannot log in as root via telnet.

- **pam_nologin.so.** You can use this module to prevent users from logging into the system. This module is also listed in the configuration file /etc/pam.d/login:

  `auth required pam_nologin.so`

  If you want to use this module, then every type of login by users can be prevented by entering the following to generate the file /etc/nologin:

  echo "No login possible because of maintenance work" > /etc/nologin

  When a user tries to log in, the message is displayed to inform the user why the login failed.

PAM Documentation Resources

The following PAM documentation is available in the directory /usr/share/doc/packages/pam/:

- **READMEs.** In the top level of this directory, there are some general README files. The subdirectory modules/ holds README files about the available PAM modules.
- **The Linux-PAM System Administrators' Guide.** This document includes everything that a system administrator should know about PAM.

  The document discusses a range of topics, from the syntax of configuration files to the security aspects of PAM. The document is available as a PDF file, in HTML format, and as plain text.
- **The Linux-PAM Module Writers' Manual.** This document summarizes the topic from the developer's point of view, with information about how to write standard-compliant PAM modules. It is available as a PDF file, in HTML format, and as plain text.
- **The Linux-PAM Application Developers' Guide.** This document includes everything needed by an application developer who wants to use the PAM libraries. It is available as a PDF file, in HTML format, and as plain text.

Thorsten Kukuk has developed a number of PAM modules for SUSE Linux and made available some information about them at http://www.suse.de/~kukuk/pam.

Exercise 2-5 Configure PAM Authentication for Digital Airlines Employees

In this exercise, you perform tests that prevent all normal users (such as geeko) from logging in to see how PAM is used by the system.

Do the following:

1. From the KDE desktop, switch to virtual console 3 (**Ctrl+Alt+F3**); then log in as **root** with a password of **novell**.
2. Create the file /etc/nologin by entering **touch /etc/nologin**.
3. Log out as root by entering **exit**.
4. Attempt to log in as **geeko** with a **N0v3ll** password.

   A Login incorrect message is displayed, indicating that you cannot log in to the system.
5. Login as **root** with a password of **novell**.
6. View the last lines of the file /var/log/messages by entering the following:

   tail /var/log/messages

   Look for the FAILED LOGIN message for geeko that indicates the failed login attempt.
7. Edit the file /etc/pam.d/login:
   a. Enter **vim /etc/pam.d/login**.
   b. Switch to the text insert mode by pressing **Insert**.
   c. Comment out the following pam_nologin.so line (add a # sign to the beginning of the line):

      #auth required pam_nologin.so

      This PAM module is required to be successful during system authentication. It checks to see if the file /etc/nologin exists, and if it does, this PAM module does not allow regular users to log in by returning a failed status.

Now that this line is commented out, PAM will not check for the file. This means that all users can log in, even if the file exists.

d. Return to the command mode by pressing **Esc**.

e. Save the file and exit vi by entering **:wq**.

8. Test the modified PAM configuration file:

   a. Log out as root by entering **exit**.

   b. Attempt to log in as **geeko** with a password of **N0v3ll**.

      You are able to log in because PAM no longer checks for the file /etc/nologin.

   c. Logout as geeko by entering **exit**.

9. Edit the file /etc/pam.d/login to uncomment the pam_nologin.so line:

   a. Log in as **root**.

   b. Enter **vim /etc/pam.d/login**.

   c. Switch to the text insert mode by pressing **Insert**.

   d. Uncomment the pam_nologin.so line (by removing the # sign) to look like the following:

      auth required pam_nologin.so

   e. Return to the command mode by pressing **Esc**.

   f. Save the file and exit vi by entering **:wq**.

   g. Log out as root by entering **exit**.

10. Try logging in as **geeko**.

    This time you receive a Login incorrect message.

11. Log in as **root**; then delete the file /etc/nologin by entering **rm /etc/nologin**.

12. Log out as root by entering **exit**.

13. Log in as **geeko**.

    Because the file /etc/nologin does not exist, login for normal users is enabled again.

14. Log out as geeko by entering **exit**.

15. Return to the desktop by pressing **Ctrl+Alt+F7**.

(End of Exercise)

Objective 6 Implement and Monitor Enterprise Security Policies

In the previous objectives, you have learned how to perform basic administrative tasks that provide secure access to the SUSE Linux environment.

However, before implementing these features in your own enterprise network environment, you need to know the following:

- Guidelines for Implementing Security Policies
- Security Rules and Tips
- SuSE Security Information Resources
- How to Monitor Login Activity

Guidelines for Implementing Security Policies

In order to provide a secure environment for your users and data in a multiuser environment, you need to consider the following guidelines when implementing your enterprise security policies on SUSE Linux Enterprise Server:

- Local Security and User Accounts
- Linux Password Encryption
- Boot Procedure Protection
- File Permissions Configuration
- Network Security and Local Security

Local Security and User Accounts

The main goal of local security is to keep users separate from each other, so no user can assume the permissions or the identity of another.

However, the user root holds the ultimate power on the system and can log in as any other local without a password and read any locally stored file.

Linux Password Encryption

On a Linux system, passwords are, of course, not stored as plain text, and the text string entered is not simply matched with the saved pattern.

If this were the case, all accounts on your system would be compromised as soon as someone got access to the corresponding file.

Instead, the stored password is encrypted and, each time it is entered, it is encrypted again with the two encrypted strings being compared.

This only provides more security if the encrypted password cannot be reverse-computed into the original text string.

This encryption is actually achieved by a special kind of algorithm, also called *trapdoor algorithm*, because it only works in one direction. An attacker who has obtained the encrypted string is not able to get your password by simply applying the same algorithm again.

Instead, it would be necessary to test all the possible character combinations until a combination is found that looks like your password when encrypted. With passwords 8 characters long, there are quite a number of possible combinations to calculate.

Boot Procedure Protection

You should configure your Linux system so it cannot be booted from a floppy disk or from CD, either by removing the drives entirely or by setting a BIOS password and configuring the BIOS to allow booting from a hard disk only.

Normally, a Linux system is started by a boot loader, allowing you to pass additional options to the booted kernel. This is crucial to your system's security.

Not only does the kernel itself run with root permissions, but it is also the first authority to grant root permissions at system start-up.

You can prevent others from using these parameters during boot by setting an additional password in /boot/grub/menu.lst.

File Permissions Configuration

As a general rule, always work with the most restrictive privileges possible for a given task.

For example, it is definitely not necessary to be the user root to read or write email. If the mail program has a bug, this bug could be exploited for an attack that acts with exactly the permissions of the program when it was started. By following the above general rule, you can minimize the possible damage.

The permissions of the more than 200,000 files included in a SUSE Linux distribution are carefully chosen. A system administrator who installs additional software or other files should take great care when doing so, especially when setting the permission bits.

Experienced and security-conscious system administrators always use the **-l** option with the command **ls** to get an extensive file list, which allows them to detect any incorrect file permissions immediately.

An incorrect file attribute not only means that files might be changed or deleted, but that the modified files might be executed by root or, in the case of configuration files, programs could use such files with the permissions of root.

This significantly increases the possibilities of an attacker. Attacks like this are called *cuckoo eggs*, because the program (the egg) is executed (hatched) by a different user (bird), just like a cuckoo tricks other birds into hatching its eggs.

A SUSE Linux system includes the files **permissions**, **permissions.easy**, **permissions.secure**, and **permissions.paranoid**, all in the directory /etc/.

The purpose of these files is to define special permissions, such as world-writable directories or, for files, the setuser ID bit (programs with the setuser ID bit set do not run with the permissions of the user that has launched it, but with the permissions of the file owner, in most cases, root).

An administrator can use the file /etc/permissions.local to add her own settings.

To learn more about the permissions topic, read the comments in the file /etc/permissions, refer to the manual page of chmod (**man chmod**), or see "Permissions and Permission Values" on 2-68.

For additional details on types of security breaches (such as buffer overflows and viruses, see Section 26.7, "Security and Confidentiality" in the SLES 9 Installation and Administration manual.

Network Security and Local Security

Network security is important for protecting from an attack that is started outside. The typical login procedure requiring a user name and a password for user authentication is still a local security issue.

In the particular case of logging in over a network, you need to differentiate between the two security aspects. What happens until the actual authentication is network security and anything that happens afterward is local security.

Security Rules and Tips

The following is a list of rules and tips you might find useful in dealing with basic security concerns:

- According to the rule of using the most restrictive set of permissions possible for every job, avoid doing your regular jobs when you are logged in as root.

  This reduces the risk of getting a cuckoo egg or a virus and protects you from your own mistakes.

- If possible, always try to use encrypted connections to work on a remote machine. Using ssh (secure shell) to replace telnet, ftp, rsh, and rlogin should be standard practice.

- Avoid using authentication methods based on IP addresses alone.

- Try to keep the most important network-related packages up-to-date and subscribe to the corresponding mailing lists to receive announcements on new versions programs such as bind, postfix, and ssh. The same should apply to software relevant to local security.

- Disable any network services you do not absolutely require for your server to work properly. This will make your system safer.

Open ports, with the socket in state LISTEN, can be found with the program **netstat**. As for the options, we recommended that you use **netstat -ap** or **netstat -anp**. The -p option lets you see which process is occupying a port under which name.

- RPM packages from SUSE are digitally signed. You can verify the integrity of any SUSE RPM package by entering the command **rpm -- checksig package.rpm**.

  The needed public gpg-key is copied to the home directory of root upon installation and can also be found on the first installation CD.

- Check your backups of user and system files regularly. Remember that if you do not test whether the backup will work, it might actually be worthless.
- Check your log files. Whenever possible, write a small script to search for suspicious entries. Admittedly, this is not exactly a trivial task. In the end, only you can know which entries are unusual and which are not.
- Use SUSEfirewall to enhance the security provided by tcpd (tcp_-wrapper).
- Design your security measures to be redundant. A message seen twice is much better than no message at all.

SuSE Security Information Resources

To handle security competently, it is important to keep up with new developments and to stay informed about the latest security issues.

One very good way to protect your systems against problems of all kinds is to install the updated packages recommended by security announcements as quickly as possible.

SUSE security announcements are published on a mailing list to which you can subscribe by using the following the link: http://www.suse.de/security.

The list suse-securityannounce@suse.de is a first-hand source of information regarding updated packages and includes members of SUSE's security team among its active contributors.

The mailing list suse-security@suse.de is a good place to discuss any security issues of interest.

The bugtraq@securityfocus.com list is one of the best-known security mailing lists worldwide. We recommend that you read this list (which receives between 15 and 20 postings per day).

You can find more information at http://www.securityfocus.com.

How to Monitor Login Activity

One of the most critical tasks you have as an administrator is to make sure that there is no suspicious activity on your system that might compromise security.

Monitoring tasks include evaluating login activity for signs of security breach such as multiple failed logins.

Reviewing files such as messages in /var/log/ also gives you information about login activity.

To monitor login activity, you can use the following commands:

- **who.** This command shows who is currently logged in to the system and information such as the time of the last login.

  You can use options such as -H (display column headings), -r (current runlevel), and -a (display information provided by most options).

For example, entering **who -H** returns information similar to the following:

```
earth: ~ # who -H
NAME      LINE      TIME            Command
root      0         Aug 23 05:41    (console)
geeko     pts/2     Aug 24 02:32    (10.0.0.50)
earth:~ #
```

- **w.** This command displays information about the users currently on the machine and their processes.

  The first line includes information the current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes.

  Below the first line is an entry for each user that displays the login name, the tty name, the remote host, login time, idle time, JCPU, PCPU, and the command line of the user's current process.

  The JCPU time is the time used by all processes attached to the tty. It does not include past background jobs, but does include currently running background jobs.

  The PCPU time is the time used by the current process, named in the What field.

  You can use options such as -h (don't display the header), -s (don't display the login time, JCPU, and PCPU), and -V (display version information).

  For example, entering **w** returns information similar to the following:

```
earth: ~ # w
USER      TTY         LOGIN@        IDLE   JCPU   PCPU  WHAT
root      0         Mon05          ?xdm?   1:48  0.02s  -0
geeko     pts/2     02:32          0.00s  0.10s  0.02s  ssh: geeko [priv]
earth:~ #
```

- **finger.** This command displays information about local and remote system users. By default, the following information is displayed about each user currently logged in to the local host:
  - Login name
  - User's full name
  - Associated terminal name
  - Idle time
  - Login time (and from where)

  You can use options such as -l (long format) and -s (short format).

  For example, entering **finger -s** returns information similar to the following:

```
earth: ~ # w
Login    Name                 Tty       Idle  Login Time       Where
geeko    The SUSE Chameleon   pts/2        -     Tue 02:32    10.0.0.50
root     root                 0          54d     Mon 05:41    console
earth:~ #
```

- **last.** This command displays a listing of the last logged in users.

  Last searches back through the file /var/log/wtmp (or the file designated by the option -f) and displays a list of all users logged in (and out) since the file was created.

  You can specify names of users and tty's to only show information for those entries.

  You can use options such as -*num* (where *num* is the number of lines to display), -a (display the hostname in the last column), and -x (display system shutdown entries and runlevel changes).

For example, entering **last -ax** returns information similar to the following:

```
earth: ~ # last -ax
geeko        pts/2     Tue Aug 24 02:32   still logged in    10.0.0.50
geeko        pts/2     Tue Aug 24 02:29 - 02:30   (00:00)    10.0.0.50
geeko        pts/2     Tue Aug 24 02:29 - 02:28   (00:07)    10.0.0.50
geeko        pts/2     Tue Aug 24 02:29 - 00:39   (00:00)    10.0.0.50
geeko        pts/2     Tue Aug 24 02:29 - 08:20   (00:07)    10.0.0.50
root         0         Mon Aug 23 05:41   still logged in    console


wtmp begins Fri Aug  6 09:04:35 2004
earth:~ #
```

- **lastlog.** This command formats and prints the contents of the last login log file (/var/log/lastlog). The login-name, port, and last login time are displays.

  Entering the command without options displays the entries sorted by numerical ID.

  You can use options such as -u *login_name* (display information for designated user only) and -h (display a one-line help message).

  If a user has never logged in the message **Never logged in** is displayed instead of the port and time.

  For example, entering **lastlog** returns information similar to the following:

```
earth: ~ # lastlog
Username         Port          From          Latest
root             0             console       Mon Aug 23 05:41:50 -0600 2004
bin                                          **Never logged in**
daemon                                       **Never logged in**
lp                                           **Never logged in**
mail                                         **Never logged in**
...
```

- **faillog.** This command formats and displays the contents of the failure log (/var/log/faillog) and maintains failure counts and limits.

  You can use options such as -u *login_name* (display information for designated user only) and -p (display in UID order).

  The command faillog only prints out users with no successful login since the last failure. To print out a user who has had a successful login since his last failure, you must explicitly request the user with the -u option.

  Entering **faillog** returns information similar to the following:

```
earth: ~ # faillog
Username       Failures   Maximum   Latest
root                  0         0   Fri Aug 20 07:20:11 -0600 2004 on tty1
geeko                 0         0   Mon Aug 16 16:48:34 -0600 2004 on tty3
earth:~ #
```

You can activate or deactivate logging activity in /var/log/lastlog and /var/log/faillog by setting options in the YaST Security settings module (see "Configure Security Settings" on 2-54.

Exercise 2-6 Change the Security Settings

As part of tightening security, SUSE provides configuration files for locking down your system. From a files perspective, there are 3 settings: easy, secure, and paranoid.

In this exercise, you change to the paranoid setting and observe the impact on the system.

Do the following:

1. From a terminal window, verify that SUID bit for ping is set by entering **ls -al /bin/ping**.

   Notice that the permissions are set to **-rwsr-xr-x**. The s in the fourth field indicates that the SUID is set, which means that any user can run ping, but it executes in memory as the owner of the file (root).

2. From the KDE desktop, select the **YaST** icon; then enter a password of **novell** and select **OK**.

3. Select **Security and Users > Security settings**.

   The Local Security Configuration dialog appears.

4. Select **Level 3 (Network Server)**; then select **Details**.

   The Password Settings dialog appears.

5. Continue by selecting **Next**.

   The Boot Settings dialog appears.

6. Continue by selecting **Next**.

   The Login Settings dialog appears.

7. Continue by selecting **Next**.

   The Adding User dialog appears.

8. Continue by selecting **Next**.

   The Miscellaneous Settings dialog appears.

9. From the Setting of File Permissions drop-down list, select **Paranoid**.

10. Configure the system for the Paranoid setting by selecting **Finish**.

11. From the terminal window, verify that SUID bit for ping is *not* set by entering **ls -al /bin/ping**.

    Notice that the permissions are now set to **-rwxr-xr-x**. The x in the fourth field indicates that the execute bit is set for root.

12. Attempt to ping DA1 by entering **/bin/ping da1**.

    You receive a ping: icmp open socket: Operation not permitted message because only root is allowed to open the socket.

13. Reset the file permissions settings to secure:

    a. From the YaST Control Center, select **Security and Users > Security settings**.

    b. Select **Level 3 (Network Server)**; then select **Details**.

       The Password Settings dialog appears.

    c. Continue by selecting **Next**.

    d. Accept the default settings for the following dialogs by selecting **Next**:

       - Boot Settings
       - Login Settings
       - Adding User

       The Miscellaneous Settings dialog appears.

    e. From the Setting of File Permissions drop-down list, select **Secure**.

    f. Configure the system for the Secure setting by selecting **Finish**.

14. From the terminal window, verify that SUID bit for ping is set by entering **ls -al /bin/ping**.

    Notice that the permissions are set back to **-rwsr-xr-x**.

15. Attempt to ping DA1 by entering **/bin/ping da1**.

    You are able to ping server DA1.

16. End the ping process by pressing **Ctrl+C**.

17. Close all open windows on the desktop.

(End of Exercise)

Summary

Objective	Summary
1. Describe Basic Linux User Security Features	One of the main characteristics of a Linux operating system is its ability to handle several users at the same time (multiuser) and to allow these users to perform several tasks on the same computer simultaneously (multitask). To maintain an environment where data and applications are secure, you learned about the following: ■ File System Security Components ■ Users and Groups ■ Ownership and Access Permissions

Objective	Summary
2. Manage Linux Users and Groups	To manage Linux user accounts groups from your SUSE Linux Enterprise Server, you learned how to do the following: ■ Create and Edit User Accounts With YaST ■ Create and Edit Groups With YaST ■ Edit User Account Properties ■ Configure Account Password Settings ■ Manage User Accounts From the Command Line ■ Manage Groups From the Command Line ■ Create Text Login Messages
3. Manage and Secure the Linux User Environment	Besides learning how to manage individual user accounts, you also learned how to do the following to manage and secure the Linux user environment: ■ Perform Administrative Tasks as root ■ Delegate Administrative Tasks With sudo ■ Set Defaults for New User Accounts ■ Configure Security Settings

Objective	Summary
4. Secure Files and Directories With Permissions	Although users might have access to navigate through the files and directories on a Linux system, you can limit what they can do with those files and directories by using access permissions. To set permissions for files and directories, you learned about the following: ■ Permissions and Permission Values ■ How to Set Permissions From the Command Line ■ How to Set Permissions From a GUI Interface ■ How to Modify Default Access Permissions ■ How to Configure Special File Permissions ■ How to Configure Additional File Attributes for ext2

Objective	Summary
5. Configure User Authentication With PAM	Linux uses PAM (Pluggable Authentication Modules) in the authentication process as a layer that communicates between users and applications. To understand how to configure PAM, you learned the following: ■ Location and Purpose of PAM Configuration Files ■ PAM Configuration File Structure ■ PAM Configuration File Examples ■ PAM Documentation Resources
6. Implement and Monitor Enterprise Security Policies	In the previous objectives, you learned how to perform basic administrative tasks that provide secure access to the SUSE Linux environment. However, before implementing these features in your own enterprise network environment, you learned the following: ■ Guidelines for Implementing Security Policies ■ Security Rules and Tips ■ SuSE Security Information Resources ■ How to Monitor Login Activity

SECTION 3 Manage the Linux File System

In this section, you discover how to manage your SUSE Linux Enterprise Server file system by learning basic tasks such as implementing partitions, creating a file system, checking the file system for errors, and performing backup and restore procedures.

Objectives

1. Select a Linux File System
2. Configure Linux File System Partitions
3. Configure a File System With Logical Volume Management (LVM)
4. Configure and Manage a Linux File System
5. Set Up and Configure Disk Quotas
6. Back Up and Restore the File System

Objective 1 Select a Linux File System

One of the key roles performed by the Linux operating system is providing storage services through creating and managing a file system.

To successfully select a file system that meets your server requirements, you need to understand the following about file systems available for Linux:

- Linux File Systems
- Linux File System Formats
- Linux File System Characteristics
- File System Journaling
- Additional File System Documentation

It is very important to keep in mind that there might be no file system that best suits all kinds of applications. Each file system has its particular strengths and weaknesses, which must be taken into account.

However, even the most sophisticated file system cannot be a substitute for a reasonable backup strategy.

For additional details on specific file systems (such as ext3 and ReiserFS), see Section 18.2 in the SLES 9 Installation and Administration manual.

Also see "Additional File System Documentation" on 3-21 at the end of this objective.

Linux File Systems

The type of file system you select depends on several factors (including speed and journaling). The following describes the file systems and formats available on Linux:

- Traditional File Systems
- Journaling File Systems
- Virtual Filesystem Switch

All of these file system types are included in the 2.6 Linux kernel (used in SUSE Linux Enterprise Server 9).

The basic Linux 2.4 kernel includes all advanced file system types. These include XFS, ReiserFS, ext3, and non-journaling file systems.

You can enter the following command to list the file system formats the kernel currently supports:

cat /proc/filesystems

Traditional File Systems

Traditional file systems supported by Linux do not journal data or metadata. These include the following:

- **ext2.** The ext2 file system is inode-based, designed for speed, is efficient, and does not fragment easily.

  Because of these features, ext2 continues to be used by many administrators, even though it does not provide a journaling feature.

  The ext2 file system has been available for many years, and is easily converted to an ext3 file system.

- **minix.** The minix file system is old and fairly limited (it was the first Linux file system), but is still sometimes used for floppy disks or RAM disks where minix's extremely low file system overhead allows for the storage of more data.
- **MS-DOS/VFAT.** FAT (File Allocation Table) is the primary file system for consumer versions of Microsoft Windows up to and including Windows Me.

  VFAT is the 32-bit version of FAT that includes long filenames.
- **HPFS.** HPFS (High Performance File System) is the native file system for IBM's OS/2 file system.

Journaling File Systems

The following file systems available for Linux include a journaling feature:

- **ext3.** ext3 is the version of the ext2 file system that supports journaling.
- **ReiserFS.** Originally designed by Hans Reiser, ReiserFS treats the entire disk partition as if it were a single database table, storing not only the file metadata, but the file itself.

  Directories, files, and file metadata are organized in an efficient data structure called a "balanced tree," which offers significant speed improvements for many applications, especially those which use lots of small files.
- **NTFS.** NTFS (New Technology File System) is the file system used by Windows NT.

  It supports multiple file systems, has file recovery for hard disk crashes, uses the Unicode character set, and provides for file names up to 255 characters long.

  Although more feature-rich than MS-DOS/VFAT, only reading of the file system is currently supported under Linux.

- **JFS.** This journaling file system from IBM was released as a production version in 2001.
- **XFS.** XFS is a high-performance journaling file system from SGI. It provides quick recovery after a crash, fast transactions, high scalability, and excellent bandwidth.

  XFS combines advanced journaling technology with full 64-bit addressing and scalable structures and algorithms.

For details on XFS, see http://oss.sgi.com/projects/xfs/.

- **Veritas's' VxFS**. VxFS is a commercial journaling file system that first shipped for Linux during 2001, and is frequently used on Unix platforms.

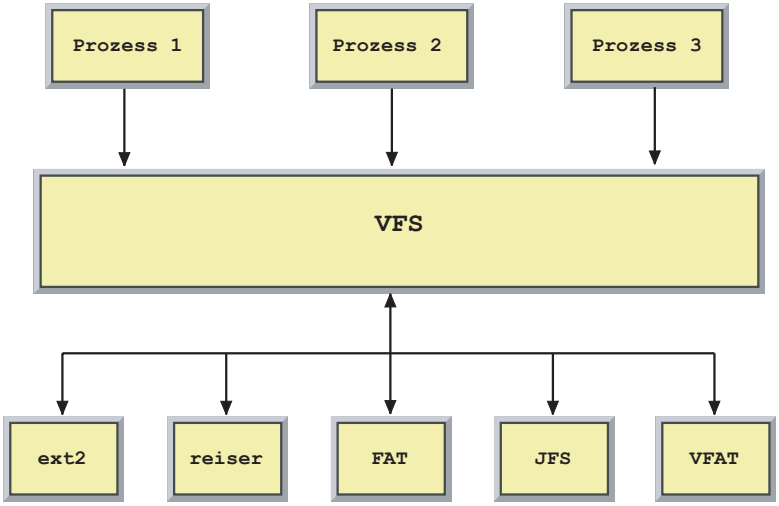
Virtual Filesystem Switch

For a user or program, it does not matter which file system format is used. The same interface to the data always appears. This is implemented by the Virtual Filesystem Switch (VFS) (also referred to as the *virtual file system*).

This is an abstract level in the kernel providing defined interfaces on the part of the processes. It includes functions such as open a file, write to a file, and read a file.

A program does not have to worry about how file access is implemented technically. The VFS forwards these requests to the corresponding driver for the file system format, as illustrated in the following:

Figure 3-1



One of the features of the VFS is to display file characteristics to the user as they are known from UNIX file system formats. This includes access permissions, even if they do not exist, as is the case with VFAT.

Linux File System Formats

File system formats in Linux are characterized by the fact that data and administration information are kept separate. Each file is described by an *inode* (index node or information node).

Each of these inodes has a size of 128 bytes and contains all the information about this file apart from the filename. This includes details such as the owner, access permissions, the size, various time details (time of modification, time of access, and time of modification of the inode), and the links to the data blocks of the file.

How data organization takes place differs from one file system format to the next. To understand the basics of file system data organization on Linux, you need to know the following:

- ext2fs File System Format
- ReiserFS Format
- Directories
- Network File System Formats

ext2fs File System Format

The ext2 file system format is, in many ways, identical to traditional UNIX file system formats. The concepts of inodes, blocks, and directories are the same.

When a file system is created (the equivalent of formatting in other operating systems), the maximum number of files that can be created is specified. This inode density (together with the capacity of the partition) determines how many inodes can be created.

Remember that it is not possible to generate additional inodes later. You can only specify an inode number when creating the file system.

An inode must exist for each file or directory on the partition. The number of inodes also determines the maximum possible number of files. Typically, an inode is generated for 4096 bytes of capacity.

On average, each file should be 4 KB in size for the capacity of the partition to be used optimally. If a large number of files are smaller than 4 KB, more inodes are used compared with the capacity.

This can result in the system preventing any more file creation, even if there is still space on the file partition.

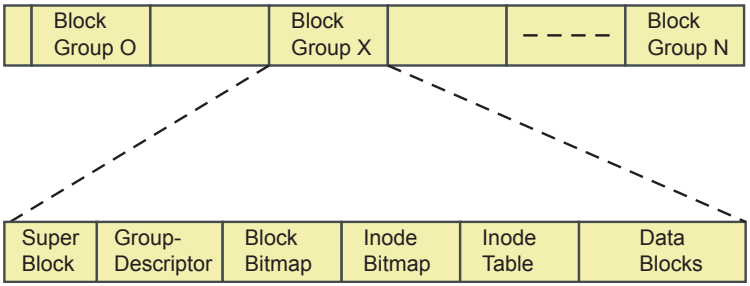
For applications that create a large number of very small files, the inode density should be increased by setting the corresponding capacity to a smaller value (such as 2048 or even 1024). However, the time needed for a file system check will increases substantially.

The space on a partition is divided into *blocks*. These have a fixed size of 1024, 2048, or 4096 bytes. You specify the block size when the file system is created; it cannot be changed later.

The block size determines how much space is reserved for a file. The larger this value is, the more space is consumed by the file, even if the actual amount of data is smaller.

In the classic file system formats (to which ext2 also belongs), data is stored in a linear chain of blocks of equal size. A specific number of blocks is grouped together in a block group (as illustrated in the following) and each block group consists of 8192 blocks:

Figure 3-2



The boot sector is located at the beginning of this chain and contains static information about the file system, including where the kernel to load can be found.

Each block group contains the following components:

- **Superblock.** The superblock is read when the file system is mounted and contains the following information about the file system:
  - The number of free and occupied blocks and inodes
  - The number of blocks and inodes for each block
  - Information about file system use, such as the time of the last mount, the last write access, and the number of mounts since the last file system check
  - A valid bit, which is set to **0** when the file system is mounted and set to **1** again by umount

    When the computer is booted, the valid bit is checked. If it is set to 0 (power failure or reset), the automatic file system repair is started.

    The remains of files that can no longer be reconstructed are stored in the directory lost+found/ (in an ext2/ext3 file system).

  For reasons of security, there are copies of the superblock. Because of this, the file system can be repaired, even if the first superblock has been destroyed.
- **Group Description.** Information is stored here about where other areas are located (such as block bitmap and inode bitmap). This information is also stored a number of times for reasons of security.
- **Block Bitmap.** Information is stored here indicating which blocks in this group are free or occupied.
- **Inode Bitmap.** Information is stored here indicating which inodes are free or occupied.

- **Inode Table.** File information is stored in this table that includes owners, access permissions, time stamps, and links to the data blocks in which the data is located.
- **Data Blocks.** This is where the actual data is located.

The ext2 file system format can process filenames with a length of up to 255 characters. With the path, a name can be a maximum of 4096 characters in length (slashes included).

A file can be up to 16 GB in size for a block size of 1024 bytes or 2 TB for a block size of 4096 bytes. The maximum file system size is 2 TB (with a block size of 1024 bytes) or 16 TB (with a block size of 4096 bytes).

The limitation on file size remains for the ext2 file system. However, the kernel can now handle files of almost any size.

ReiserFS Format

On a file system with ext2 and a block size of 1024 bytes, a file 8195 bytes in size occupies 8 blocks completely and a ninth block with 3 bytes.

Even though only 3 bytes are occupied, the block is no longer available. This means that approximately 11 percent of available space is wasted.

If the file is 1025 bytes in size, 2 blocks are required, 1 of which is almost completely empty. Almost 50 percent of the space is wasted.

A worst case occurs if the file is very small: even if the file is only 50 bytes in size, a whole block is used (95 percent wasted).

A solution to this problem is provided by the ReiserFS format, which organizes data in a different way. This file system format has (currently) only 1 fixed block size of 4096 bytes.

However, small files are stored more efficiently. Only as much space is reserved as is actually required—not an entire block. Small files or the ends of files are stored together in the same block.

The inodes required are not generated when the file system is created, but only when they are actually needed. This allows a more flexible solution to storage requirements, increasing efficiency in the use of hard drive space.

Another advantage of the ReiserFS is that access to files is quicker. This is done through the use of balanced binary trees in the organization of data blocks.

However, balanced trees require considerably more processing power because after every file is written the entire tree must be rebalanced.

The current version of the ReiserFS (3.6) contained in the kernel from version 2.4 on allows a maximum partition size of 16 TB. A file also has a maximum size of 16 TB.

The same limitations exist for filenames as with the ext2 file system format.

Directories

Inodes contain all the administrative information for a file, but not the filename. This is where directories are useful.

Like a catalog, directories contain information on other files. This information includes the number of the inode for the file and its name.

Directories serve as a table in which inode numbers are assigned line-by-line to filenames. You can view the inode assigned to a filename by using the command **ls -i**, as in the following:

```
DA50:~ # ls -i /
       2 .      104002 cdrom    80045 floppy   104081 mnt  103782 sbin
       2 ..      99068 dev      95657 home      81652 opt   80044 tmp
  104005 bin    104004 dvd     102562 lib           1 proc      4 usr
       2 boot    95722 etc      95718 media     81598 root  80046 var
```

Each filename is preceded by the inode number.

On this particular SUSE Linux server there are 2 partitions: one configured on the root directory /, and one configured on the directory /boot/.

Because inodes are always uniquely defined on one partition only, the same inode numbers exist on each partition (at least in part).

In the example, the 2 entries "." (a link to the current directory—here the root directory) and boot (the second partition is mounted on this directory) have the same inode number (2), but they are located on different partitions.

The file "..", which is actually a link to the previous layer in the direction of the root directory, also has an inode number of 2.

Because you are already in the root directory, this link points to itself. It is another name entry for an inode number.

The table (the directory file) for the root directory can be represented as in the following example:

Table 3-1

Inode Number	Filename
1	proc
2	
2	.
2	boot
4	usr
80044	tmp
80045	floppy
80046	var
81598	root
81652	opt
95718	media
95722	etc
99068	dev
99657	home
102562	ib
103782	sbin
104002	cdrom
104004	dvd
104005	bin
104081	mnt

Network File System Formats

In addition to the already mentioned file system formats on the local computer, Linux also understands various network file system formats. The most significant of these is the Network File System (NFS), the standard in the UNIX world.

With NFS, it does not matter which file system format is used locally on individual partitions. As soon as a computer is functioning as an NFS server, it provides its file systems in a defined format NFS clients can access.

Using additional services included on SUSE Linux Enterprise Server, Linux can also work with the network file system formats of other operating systems.

These include the Server Message Block (SMB) format used in Windows and the Netware Core Protocol (NCP) from Novell.

SMB allows Linux to mount Windows 9*x*/NT network shares.

Linux File System Characteristics

Linux and UNIX file systems all start from a root (/) directory and include all other physical or network file systems under the root directory.

For example, if you insert a CD or DVD into your Linux computer, it is mounted at a specific directory under root (such as /media/cdrecorder or /media/cdrom). You can then view the contents of the CD or DVD by simply changing to that directory.

This approach is unlike Windows and other operating systems where you access physical or network file systems using drive mapping (such as drive D: for a CD or DVD).

Another difference between Linux and a file system such as Windows is that filenames are case sensitive.For example, the filename **steps** is different from **STEPS** and **Steps** or **sTEPs**.

If a filename appears as STEPS, you need to enter **STEPS** at the command line or Linux will not recognize the filename.

Long filenames are supported by native Linux file systems (such as ext2 and ReiserFS) and these filenames are case sensitive.

Linux also includes support for timestamps such as ctime (inode modification time), mtime (modification time), and atime (access time).

In addition to these characteristics, you also need to know the following:

- Linux File Types
- Linux File System Directories

Linux File Types

Normal files and directories in Linux are also known to other operating systems. But there are 4 additional types of files that are UNIX-specific.

The following describes these file types, along with normal files and directories available in Linux:

- **Normal files.** Normal files refer to files as they are also known to other operating systems: a set of contiguous data addressed with 1 name. These include files such as ASCII texts, executable programs, and graphics files.

  With some limitations, you can chose the filenames. There is no association between filename and file type (for example, report.txt).

A number of filenames still retain this structure, but these are requirements of the corresponding applications, such as a word processing program or a compiler.

- **Directories.** Directories are special files containing information about other files. In particular, they contain 2 entries that implement the structure of the hierarchical file system.

  One of these entries (“.”) points to the directory itself. The other entry (“..”) points to the entry one level higher in the hierarchy.

- **Device files.** Each piece of hardware in a Linux system is represented by a device file (except for network cards). The files represent links between the hardware components or the device drivers in the kernel and the applications.

  Every program that needs to access hardware must do so through the corresponding device file. The programs write to or read from a device file. The kernel makes sure that the data finds its way to the hardware or can be read from there.

- **Links.** Links are references to files located at other points in the file system.

  Through the use of links, maintaining data becomes much simpler. Changes only need to be made to the original file. They are then automatically valid for all links.

- **Sockets.** Sockets refer to special files with which data exchange between two locally running processes can be implemented through the file system.

- **FIFOs.** *FIFO* (first in first out) or *named pipe* is a term used for files to exchange data between processes. However, they can only exchange data in 1 direction.

Linux File System Directories

The Linux file system is hierarchical, much like other operating systems you might have used. The root of the file system is named /, with the following subdirectories:

Figure 3-3

/ bin
/ boot
/ dev
/ etc
/ home
/ lib
/ media
/ mnt
/ opt
/ proc
/ root
/ sbin
/ srv
/ suse_linux
/ sys
/ tmp
/ usr
/ var

Directories below / are referenced from the root. For example, if you have a directory named bin that is located in a directory named usr that is located right below the root directory, it would be referenced as /usr/bin/.

The directories used in a Linux system are defined by the File System Hierarchy standard (FHS) in the Linux Standards Base (LSB).

The FHS specifies which directories must be located on the first level after the root directory and what they contain, but allows for flexibility in defining your own hierarchy in other areas.

The FHS defines a two-layered hierarchy:

- The directories in the top layer (under the root directory “/”)

- The directories under /usr/.

For additional information on the File System Hierarchy, see http://www.pathname.com/fhs/.

Some of the important default Linux directories include the following:

- **/etc/.** This directory contains configuration files.
- **/dev/.** This directory contains special link files that reference hardware in the system. For example, /dev/fd0 references floppy disk A. The first IDE hard disk is referenced by /dev/hda.

  It also contains special links for removable devices. For example, the CD-ROM drive is referenced by /dev/cdrom; /dev/fd0 (floppy drive) is linked to /dev/floppy.
- **/usr/.** This directory contains program files.
- **/var/.** This directory contains data such as print spool directories, mail storage, log files, and other temporary files.
- **/tmp/.** This directory stores temporary files created by running applications.
- **/home/.** This directory contains user home directories.
- **/root/.** This is the root user's home directory.
- **/bin/.** This directory contains essential command-line utilities such as vi, rpm, ls, mkdir, more, mv, grep, and tar.
- **/sbin/.** This directory contains essential system executables such as fsck, grub, mkfs, arp, fdisk, and ifconfig.
- **/mnt/.** This directory is used to mount devices or remote file systems using Samba or NFS. On some Linux distributions (such as Red Hat) it is also used to mount removable devices.

- **/media/.** This directory is used to mount removable devices on some Linux distributions (such as SUSE Linux).

File System Journaling

To make an appropriate decision about which file system to choose for which partition, you need to understand the following about Linux file systems and journaling:

- Journaling and File System Transactions
- ext2 and ext3 Comparison

Journaling and File System Transactions

File systems are basically databases that store files and use file information such as the filename and timestamp (called *metadata*) to organize and locate the files on a disk.

When you modify a file, the file system performs the following transactions:

- It updates the file (the data)
- It updates the file metadata

Because there are two separate transactions, corruption can happen when only the file data is updated (but not the metadata) or vice versa, resulting in a difference between the data and metadata.

This can happen when a large file is being downloaded from the Web and there is an interruption (such as a power outage). The data might be written first, but the metadata might not be updated.

When there is a difference between the data and metadata, the state of the file system is inconsistent and requires repair.

Although traditional file systems (such as ext2) require you to use the command fsck in a UNIX or Linux environment to check and repair the file system, you do not need to check and repair journal-based file systems such as ReiserFS.

In a journal-based file system, the journal keeps a record of all current transactions, and updates the journal as transactions are completed.

For example, when you first start copying a file from a network server to your workstation, the journaled file system submits an entry to the journal indicating that a new file on the workstation is being created.

After the file data and metadata are copied to the workstation, an entry is made indicating that the file was created successfully.

While recording entries in a journal requires extra time for creating files, it makes recovering an incomplete transaction easy as the journal can be used to repair the file system.

ext2 and ext3 Comparison

The Linux file system that been the standard file system for a long time is version 2 of the extended file system (*ext2fs* or *ext2*).

The latest version, ext3fs, has been further developed to provide journaling functionality. Logs of files that are in use or opened are continually updated. If a system crash occurs, a check of the individual partitions is made automatically.

With ext2, the complete partition (all files) must be checked, because no information is available as to whether a file was open at the time of the crash.

With ext3, only the open files need to be checked. This speeds up the system check and any repairs that might be required, so the system is available more quickly.

In addition, ext3 lets you journal all file data and metadata or simply journal the metadata.

You normally use a nonjournaling system such as ext2 on a small partition (such as a /boot partition) because the administrative information overhead of a system such as ReiserFS is much larger.

Additional File System Documentation

Each of the Linux file systems maintains its own home page on which to find mailing list information, further documentation, and FAQs. These include the following:

Table 3-2

File System	URL
ext2	http://e2fsprogs.sourceforge.net/
ext3	http://olstrans.sourceforge.net/release/OLS2000-ext3/OLS2000-ext3.html
ReiserFS and Reiser4	http://www.namesys.com/
IBM's JFS	http://www-128.ibm.com/developerworks/linux/library/l-jfs.html
SGI's XFS	http://oss.sgi.com/projects/xfs/
Veritas's' VxFS	http://www.veritas.com/us/products/filesystem/

A comprehensive multipart tutorial about Linux file systems can be found at IBM developerWorks at the following URL:

http://www-106.ibm.com/developerworks/library/l-fs.html

For a comparison of the different journaling file systems in Linux, look at Juan I. Santos Florido's article at Linuxgazette:http://www.linuxgazette.com/issue55/florido.html.

If you are interested in an in-depth analysis of Large File Support (files larger than 2 GB) in Linux, visit Andreas Jaeger's LFS site at http://www.suse.de/~aj/linux_lfs.html.

Objective 2 Configure Linux File System Partitions

A basic task of all system administrators is maintaining file system layouts. Under Linux (and UNIX), new partitions can be transparently grafted into existing file system structures using the **mount** command.

In most cases, YaST proposes a reasonable partitioning schema during installation that can be accepted without change. However, you can also use YaST to customize partitioning after installation.

To implement partitions on your SUSE Linux Enterprise Server, you need to know the following:

- Partition Types
- Linux Device and Partition Names
- Design Guidelines for Implementing Partitions
- Design Guidelines for Optimizing Partitions
- How to Manage Partitions With YaST

Partition Types

Every hard disk (on an Intel platform) has a partition table with space for 4 entries. An entry in the partition table can correspond to a primary partition or an extended partition. However, only one extended partition entry is allowed.

A *primary partition* consists of a continuous range of cylinders (physical disk areas) assigned to a particular operating system. If you only had primary partitions, you would be limited to 4 because of the 4 entry limitation in the partition table.

Extended partitions are also continuous ranges of disk cylinders, but an extended partition can be subdivided into *logical partitions*. Logical partitions do not require entries in the partition table.

In other words, an extended partition is a container for logical partitions.

Because the extended partition should span the entire remaining free cylinder range, configure your primary partitions first before configuring the extended partition.

After configuring the extended partition, create multiple logical partitions within the extended partition. The maximum number of logical partitions is 15 on SCSI disks and 63 on (E)IDE disks.

Linux Device and Partition Names

The following table shows the names of Linux devices:

Table 3-3

Device	Linux Name
Primary master IDE hard disk	/dev/hda
Primary slave IDE hard disk	/dev/hdb
Secondary master IDE hard disk	/dev/hdc
Secondary slave IDE hard disk	/dev/hdd
First SCSI hard disk	/dev/sda
Second SCSI hard disk	/dev/sdb
Third SCSI hard disk	/dev/sdc
Fourth SCSI hard disk	/dev/sdd

Partitions follow the naming convention of the device name and partition number.

For example, the first partition on the first IDE drive would be /dev/hda1 (/dev/hda + 1 as the first partition). The first logical partition defined on an IDE hard disk will always be number 5.

The following table shows the partition names corresponding to the device the partition is defined on:

Table 3-4

Partition	Linux Name
First partition on first IDE hard drive	/dev/hda1
Second partition on first IDE hard drive	/dev/hda2
First partition on first SCSI hard drive	/dev/sda1
First logical partition on first IDE hard drive	/dev/hda5
Second logical partition on first IDE hard drive	/dev/hda6

For example, if you perform a new installation of SuSE Linux on a system with 2 IDE drives you might want the first drive to include a partition for swap and /. You might want to put all logs, mail, and home directories on the second hard drive.

The following is an example of how you might want to partition the disks (it assumes that the CD-ROM drive is the slave on the first IDE controller):

Table 3-5

Partition	Linux Name
Swap partition	/dev/hda1
/ partition	/dev/hda3
Extended partition on second disk	/dev/hdc1
/var as a logical partition on second disk	/dev/hdc5
/home as a logical partition on second disk	/dev/hdc6
/app1 as a logical partition on second disk	/dev/hdc7

On older installations you often find a small partition for /boot/. The reason for this is that the boot loader LILO needed the kernel within the first 1024 cylinders of the hard disk to boot the machine.

Design Guidelines for Implementing Partitions

YaST normally proposes a reasonable partitioning scheme with sufficient disk space. This is usually a swap partition (between 256 and 500 MB) and with the rest of the disk space reserved for a / partition.

In addition, if there is an existing partition on the hard drive, YaST attempts to maintain that partition.

If you want to implement your own partitioning scheme, consider the following recommendations:

- **File system size.** The following are recommendations for different system types:
  - **Minimal System: 700 MB.** No graphical interface (X Window System) is installed, which means that only console applications can be used. Also, only a very basic selection of software is installed.
  - **Minimal System with Graphical Interface: 1 GB.** This includes the X Window System and some applications.
  - **Default System: 1.5 GB.** This includes a modern desktop environment, like KDE or GNOME, and also provides enough space for large application suites like Netscape or Mozilla.
  - **Full Installation: 2.5 GB.** All the packages included with SUSE Linux can be installed.

- **Disk space distribution.** Depending on the amount of space and how the computer will be used, adjust the distribution of the available disk space. The following are some basic guidelines:
  - **Up to 4 GB.** One partition for the swap space and one root partition (/). In this case, the root partition must allow for those directories that often reside on their own partitions if more space is available.
  - **4 GB or more.** A swap partition, a root partition (1 GB), and 1 partition each for the following directories as needed: /usr/ (4 GB or more), /opt/ (4 GB or more), and /var/ (1 GB). The rest of the available space can be used for /home/.
- **Boot partition.** Depending on the hardware, it might also be useful to create a boot partition (/boot) to hold the boot mechanism and the Linux kernel.

  This partition should be located at the start of the disk and should be at least 8 MB or 1 cylinder.

  As a rule of thumb, always create such a partition if it was included in YaST's original proposal. If you are unsure about this, create a boot partition to be on the safe side.
- **Software and /opt/.** Some (mostly commercial) programs install their data in /opt/. In this case, you might want to create a separate partition for /opt/ or make the root partition large enough. KDE and GNOME are also installed in /opt/.
- **Additional partitions.** If the partitioning is performed by YaST and other partitions are detected in the system, these partitions are also entered in the file /etc/fstab to enable easy access to this data.

  The following is an example:

  ```
  /dev/sda1 /data1 auto noauto,user 0 0
  /dev/sda8 /data2 auto noauto,user 0 0
  ```

- **Executable files.** Partitions, whether they are Linux or FAT, are specified by YaST with the options **noauto** and **user**. This allows any user to mount or unmount these partitions as needed.

  For security reasons, YaST does not automatically enter the **exec** option, which is needed for executing programs from the respective location. However, you can enter this option manually.

  Entering the exec option is necessary if you encounter system messages such as bad interpreter or Permission denied.

Design Guidelines for Optimizing Partitions

The following guidelines are provided if you want to optimize a system for security and speed and are prepared to reinstall the entire existing system if necessary.

First, consider the following questions:

- How will the computer be used (file server, application server, compute server, standalone machine)?
- How many people will work with this computer (concurrent logins)?
- How many hard disks are installed? What is their size and type (EIDE, SCSI, or RAID controllers)?

Consider the following guidelines and suggestions:

- Size of the Swap Partition
- Processor Speed and Main Memory Size
- Standalone Computer Guidelines
- File Server Guidelines
- Computer Server Guidelines

Size of the Swap Partition

Many sources state the rule that the swap size should be at least twice the size of the main memory. This is an outdated guideline from the past when 8 MB RAM was considered a considerable amount of memory.

In the past, the aim was to equip the machine with about 30 to 40 MB of memory (RAM plus swap space). Modern applications require even more memory. For normal users, 256 MB of swap partition space is currently a reasonable value.

Never configure your system without any swap partition space.

Processor Speed and Main Memory Size

In Linux, the size of main memory is often more important than the processor speed. One reason for this is the ability of Linux to create dynamic buffers containing hard disk data.

For this purpose, Linux uses various techniques, such as read ahead (reading sectors in advance) and delayed write (postponing and bundling write access). Delayed write is the reason why you should not simply switch off your Linux machine.

Read ahead and delayed write contribute to the fact that the main memory seems to fill up over time and that Linux is so fast.

For additional details, see Section 12.2.6 in the SLES 9 Installation and Administration manual.

Standalone Computer Guidelines

As Linux becomes more popular, it is being used as a standalone system. The following provides estimated disk space requirements for home or office use:

Table 3-6

Installation	Required Disk Space
Minimum	500 MB to 750 MB
Small	750 MB to 3 GB
Medium	3 GB to 10 GB
Large	More than 10 GB

The following are examples for typical workstation installations:

- **Standard workstation (wmall)**

  To install Linux on a 2.5 GB hard disk, use 256 MB for the swap partition and the rest for / (root partition).

- **Standard workstation (average)**

  If you have 8 GB available for Linux, use 256 MB for swap, 2.5 GB for /, and the rest for a separate /home partition.

- **Standard workstation (deluxe)**

  If you have more than 10 GB available, there is no standard way to partition the disk.

File Server Guidelines

Hard disk performance is crucial on a file server. Use SCSI devices if possible. Keep in mind the performance of the disk and the controller.

Optimizing the hard disk access is vital for file servers in networks of more than 20 users.

For example, if you want to set up a Linux file server for the home directories of 25 users, and if the average user requires 2 GB for personal data, a 50 GB partition mounted under /home is probably sufficient.

For 50 users, you would need 100 GB. If possible, split /home into two 50 GB hard disks that share the load (and access time).

Web browser caches should be stored on local hard disks.

Computer Server Guidelines

A computer server is generally a powerful machine that carries out extensive calculations in the network. Such a computer is normally equipped with a large main memory (more than 512 RAM). Fast disk throughput is only needed for the swap partitions. If possible, distribute swap partitions to multiple hard disks.

How to Manage Partitions With YaST

You can use the YaST Expert Partitioner during or after installation to customize the default or existing partition configuration.

In this part, you learn how to do the following:

- Create a Partition
- Edit a Partition
- Resize a Partition

For details on all the options available in the Expert Partitioner, see Section 1.7.5 in the SLES 9 Installation and Administration manual.

Create a Partition

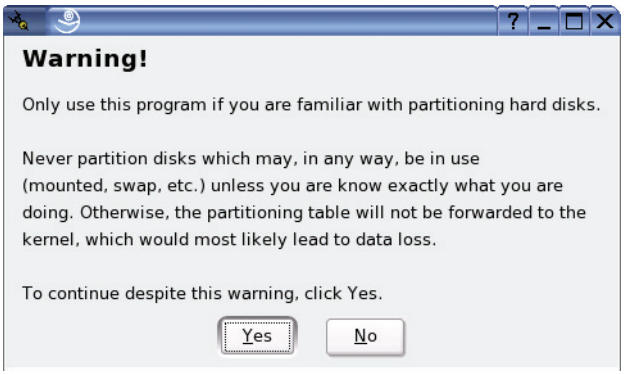
The following are the basic steps for accessing the Expert Partitioner after installation to create a partition:

1. From the KDE desktop, start the YaST Expert Partitioner module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > Partitioner**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 disk**.

   The following appears:

Figure 3-4

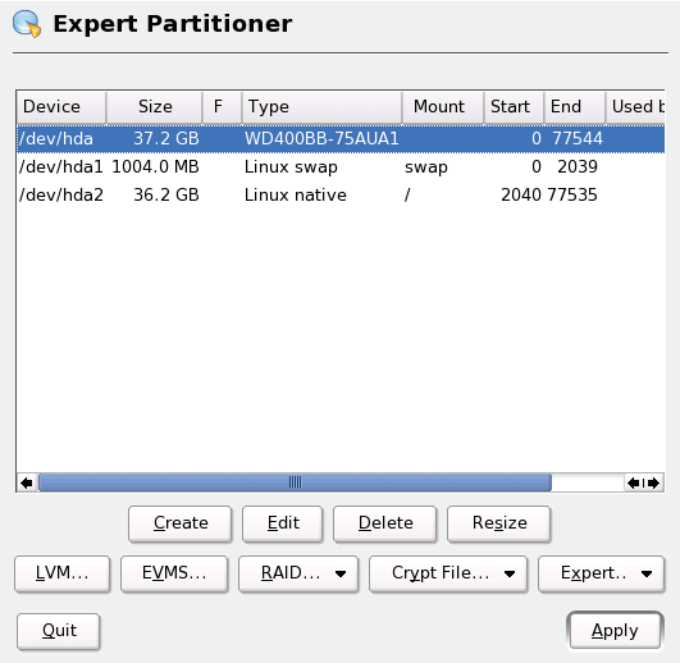


   This warning gives you the opportunity to make sure you know what you are doing and the impact (data loss) it can have on your system.

2. Continue by selecting **Yes**.

The expert partitioner appears:

Figure 3-5



The expert partitioner lets you manually modify the partitioning of your hard disk. You can manage the list of partitions by adding (**Create**), editing (**Edit**), deleting (**Delete**), or resizing (**Resize**) partitions.

All existing or suggested partitions on all connected hard disks are displayed in the partition list. Entire hard disks are listed as devices without numbers (such as /dev/hda or /dev/sda). Partitions are listed as parts of these devices (such as /dev/hda1 or /dev/sda1).

The size, type, file system, and mount point of the hard disks and their partitions are also displayed. Any free hard disk space is also listed.

The mount point describes where the partition is mounted in the Linux file system tree.

To provide more disk space to Linux, you can free the needed space starting from the bottom toward the top of the list (starting from the last partition of a hard disk toward the first).

3. Create a partition by selecting **Create**.

   If several hard disks are installed, a dialog appears that lets you select a hard disk.

4. (Conditional) Select the hard disk for creating the partition.

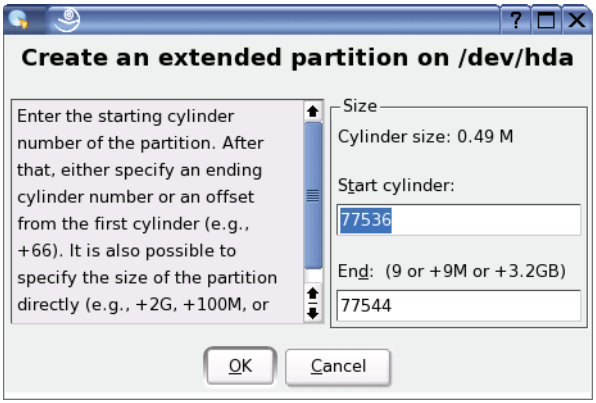
   The following appears:

Figure 3-6

5. Select the type of partition you want to create (**Primary** or **Extended**); then select **OK**.

If you select an extended partition, the following appears:

Figure 3-7



6. In the **Start cylinder** field, enter (or accept the default) beginning cylinder.
7. In the **End** field, enter one of the following:
   - The number of the ending cylinder (such as **77544**)

     or
   - The offset from the first cylinder (such as **+66** cylinders)

     or
   - A specific partition size (such as **+2G** or **+100M**)
8. When you finish indicating a starting cylinder and partition size, continue by selecting **OK**.

   If the value in the End field is not valid, a warning message appears.
9. (Conditional) Have YaST enter the maximum value for you by selecting **Yes**, or enter a value by selecting **No**.
10. Continue by selecting **OK**.

    You are returned to the Expert Partitioner with the new partition displayed in the partitions list.

Edit a Partition

After creating a partition, you can edit the partition parameters (except for start cylinder and size) with the YaST Expert Partitioner by doing the following:

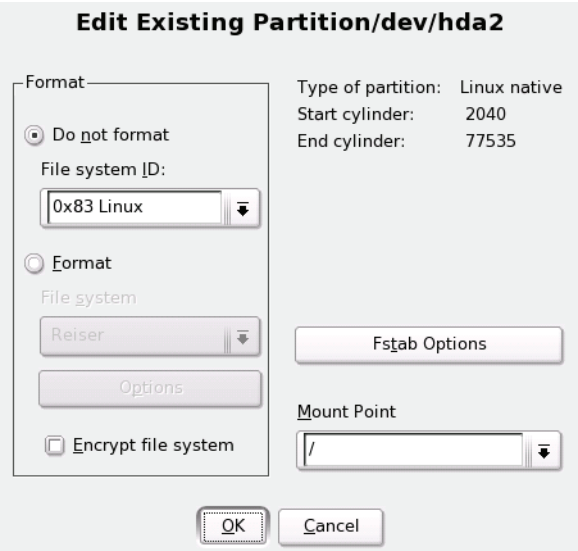
1. From the KDE desktop, start the YaST Expert Partitioner module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > Partitioner**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 disk**.

   The Expert Partitioner appears.
2. Select the ***partition*** you want to edit; then select **Edit**.

   The following appears:

Figure 3-8



3. From this dialog, you can configure the following options:
   - **File system ID.** If you do not want to format the partition, select **Do not format**, and then ensure that the partition is registered correctly by selecting a ***file system ID*** from the drop-down list.
   - **File System.** To format the partition, select **Format**, and then select a file system (such as **Ext3** or **Reiser**) for the partition from the drop-down list.
   - **Options.** If you select **Format**, you can set various options, depending on the selected file system, by selecting **Options**.
   - **Encrypt file system.** Select this option to have all data written to the hard disk in encrypted form.
   - **Fstab Options.** Select this option to set various file system parameters in /etc/fstab. These include mounting parameters and the data journaling mode.
   - **Mount Point.** Enter the directory where the partition should be mounted in the file system tree.
4. When you finish editing the options, continue by selecting **OK**.

   A warning message appears, cautioning you about the changes you've made.
5. Save the changes by selecting **OK**.

   You are returned to the Expert Partitioner dialog.

Resize a Partition

After creating a partition, you can resize the partition (if free space is available) with the YaST Expert Partitioner.

If free space is not available, you need to use a utility such as Partition Magic to resize the partition.

Do the following:

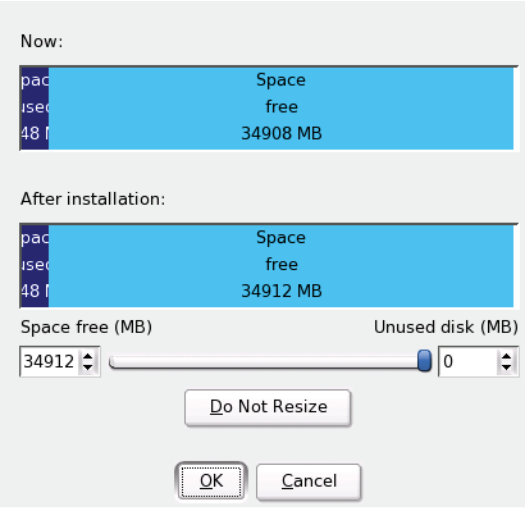
1. From the KDE desktop, start the YaST Expert Partitioner module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > Partitioner**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 disk**.

   The Expert Partitioner appears.
2. Select the ***partition*** you want to resize; then select **Edit**.

   The following appears:

Figure 3-9



3. Resize the partition by doing one of the following:
   - Increase or decrease the free space on the partition by entering a value in the **Space free** field.

     or

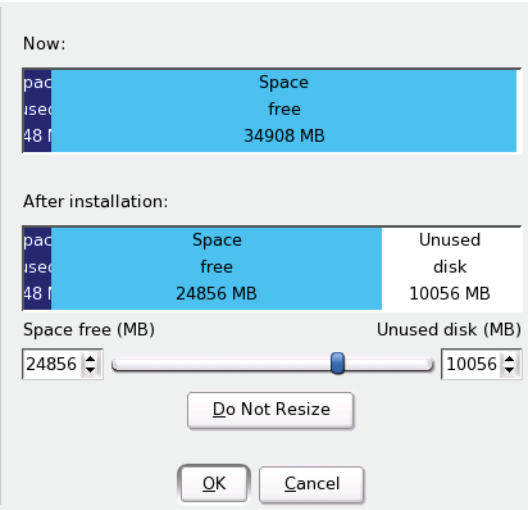
- Increase or decrease the unused disk space on the hard disk by entering a value in the **Unused disk** field.

  or

- Use the slider bar to adjust the free space/unused space ratio for the partition.

The results of the new configuration appear in the lower bar graph:

Figure 3-10



4. (Optional) If you want to reset the partition size to its original setting, select **Do Not Resize**; then select **OK**.

5. Accept the new partition size by selecting **OK**.

   If the partition is currently mounted, a warning message appears indicating that you cannot resize the partition until you unmount the file system.

6. (Conditional) Close the dialog by selecting **OK**; then unmount the partition and try resizing it again.

   You are returned to the Expert Partitioner.

At this point, the partition size has not been changed.

7. Change the partition size by selecting **Apply**.

Exercise 3-1 Configure Partitions on Your Hard Drive

In this exercise you do the following:

- Part I: Use YaST to Create a Partition and File System
- Part II: Manually Partition With fdisk

Part I: Use YaST to Create a Partition and File System

In this part of the exercise you create partitions as a prerequisite step to adding new file systems.

Do the following:

1. Create a new ext2 partition with YaST:
   a. From your KDE desktop, select the **YaST** icon; then enter a password of **novell** and select **OK**.

      The YaST Control Center appears.
   b. Select **System > Partitioner**.

      A warning message appears.
   c. Continue by selecting **Yes**.

      After a few moments the Expert Partitioner dialog appears.
   d. Create a new partition by selecting **Create**.
   e. Create a primary partition by making sure **Primary partition** is selected; then select **OK**.

      A Create a primary partition dialog appears.
   f. Configure the new primary partition by entering or selecting the following:
      - File system: **Ext2**
      - End (cylinder): **+500M**
      - Mount Point: **/apps**

g. When you finish, confirm the partition definition by selecting **OK**.

   You are returned to the Expert Partitioner dialog where the new partition is added to the list.

h. Add the new partition to the hard drive by selecting **Apply**.

   A dialog appears asking if you really want to execute the changes.

i. Continue by selecting **Finish**.

2. Verify creation of the new partition:

a. Open a terminal window; then su to root (**su -**) with a password of **novell**.

b. Verify that the new partition is mounted by entering **mount**.

   You should see the following line:

```
/dev/hda3 on /apps type ext2 (rw)
```

c. Verify that the appropriate entry was added to the /etc/fstab for the new partition by entering the following:

   cat /etc/fstab

   You should see the following:

```
/dev/hda3    /apps    ext2    acl,user_xattr  1 2
```

   This entry makes sure that when the system boots, the new file system is mounted.

3. Unmount the file system by entering **umount /apps**.

4. Verify that the file system is no longer mounted by entering **mount**.

The /dev/hda3 partition is not displayed.

5. Start a file system check on hda3 running in verbose mode with an automatic response of yes to prompts by entering the following:

e2fsck -f -y -v /dev/hda3

Check for any bad blocks.

6. Mount the /apps file system again by entering **mount /apps**.
7. Verify that the file system mounted by entering **mount**.

Part II: Manually Partition With fdisk

Do the following:

1. From the command line, start the utility fdisk on the first IDE hard disk on your server by entering **fdisk /dev/hda**.
2. View the current partition table in fdisk by entering **p**.

   Notice that there are 3 partitions (hda1, hda2, and hda3).
3. Create a new extended partition that uses the remaining space on the disk by entering **n** and then **e**; then press **Enter** twice.
4. View the updated partition table in fdisk by entering **p**.

   Notice that a new hda4 partition has been added to the table.
5. Create a new 500MB Win95 FAT32 logical partition as the first partition in the extended partition by doing the following:
   a. Create a new partition by entering **n**.
   b. Accept the default first cylinder by pressing **Enter**.
   c. Indicate the partition size by entering **+500M**.
   d. Change the partition type to Win95 FAT32 by entering **t**, **5**, and then **b**.
   e. Verify the new partition configuration by entering **p**.

      Notice that a new hda5 partition has been added to the table.
6. Create 2 new logical partitions with the partition type of Linux (the default) by doing the following:
   a. Create a new partition by entering **n**; then accept the default first cylinder by pressing **Enter**.
   b. Indicate the partition size by entering **+1G**.

c. Create a new partition by entering **n**; then accept the default first cylinder by pressing **Enter**.

d. Indicate the partition size by entering **+2G**.

e. Verify the new partition configuration by entering **p**.

Notice that 2 new partitions (hda6 and hda7) have been added to the partition table.

7. Write the new partition table to your hard drive and exit fdisk by entering **w**.

8. Reboot the system by entering **reboot**.

   It is important to reboot after repartitioning, especially with IDE devices. SCSI disk changes are much more forgiving.

9. When the server reboots, log in as **geeko** with a password of **N0v3ll**.

(End of Exercise)

Objective 3 Configure a File System With Logical Volume Management (LVM)

Logical volume management (LVM) provides a higher-level view of the disk storage on a computer system than the traditional view of disks and partitions. This gives you much more flexibility in allocating storage to applications and users.

When you create logical volumes with LVM, you can resize and move logical volumes while partitions are still mounted and running.

You can also use LVM to manage logical volumes with names that make sense (such as “development” and “sales”) instead of physical disk names such as “sda” and “sdb.”

To configure a file system with LVM, you need to know the following:

- LVM Components
- LVM Features
- How to Configure Logical Volumes With YaST

Starting from kernel version 2.6, you can use LVM version 2, which is backward-compatible with the previous LVM and enables the continued management of old volume groups.

Instead of LVM2, you can also use EVMS (Enterprise Volume Management System), which offers a uniform interface for logical volumes as well as RAID volumes. Like LVM2, EVMS makes use of the device mapper in kernel 2.6.

LVM Components

Conventional partitioning of hard disks on a Linux file system is basically inflexible. When a partition is full, you have to move the data to another medium before you can resize the partition, create a new file system, and copy the files back.

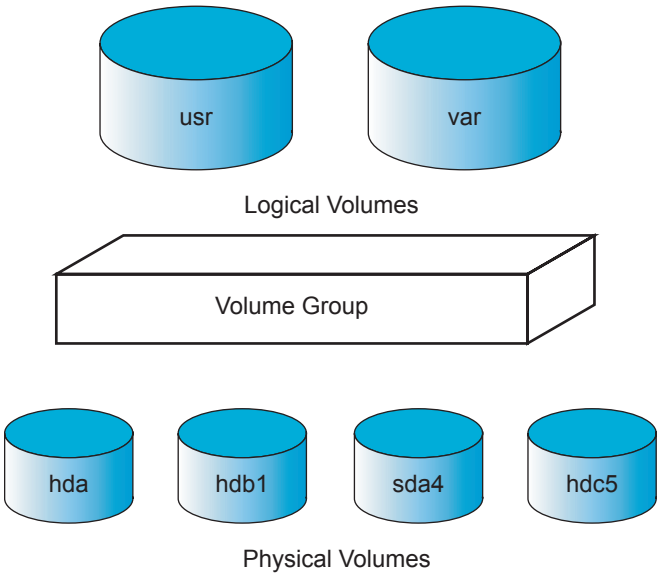
Normally, these changes cannot be implemented without changing adjacent partitions, whose contents also need to be backed up to other media and written to their original locations after the repartitioning.

Because it is difficult to modify partitions on a running system, LVM was developed. It provides a virtual pool of memory space (called a *volume group*) from which logical volumes can be generated if needed. The operating system accesses these instead of the physical partitions.

This approach lets you resize the physical media during operation without affecting the applications.

The basic structure of LVM includes the following components:

Figure 3-11



Several *physical volumes* (entire hard disks or individual partitions) are combined to a super unit referred to the *volume group*. You can add hard disks or partitions to the volume group during operation whenever necessary (no unmounting required).

The volume group can also be reduced in size by removing hard disks or partitions. The volume group, in turn, can be split into several logical volumes that can be addressed with their device names (such as /dev/system/usr) like conventional partitions with file systems.

Just as with other direct manipulations of the file system, a data backup should be made before configuring LVM.

LVM Features

Implementing LVM makes sense for heavily used home computers as well as enterprise servers. LVM also makes sense for large data collections such as databases, MP3 archives, or user directories.

The following are features of LVM that help you implement storage solutions:

- You can combine several hard disks or partitions into a large volume group.
- Provided the configuration is suitable, you can enlarge a logical volume (such as /usr) when free space is exhausted.
- You can add hard disks to the volume group in a running system, providing you have hot-swappable hardware capable of such actions.
- You can add logical volumes in a running system, providing there is free space in the volume group.
- You can use several hard disks with improved performance in the RAID 0 (striping) mode.
- You can add up to 256 logical volumes.
- The Snapshot feature enables consistent backups (especially for servers) in the running system.

How to Configure Logical Volumes With YaST

The following are the basic steps for configuring logical volumes (LVM) with YaST:

1. Define the LVM partition (physical volume) on the hard drive:

   During (or after) the installation of SUSE Linux Enterprise Server, you need to configure the LVM partition on the hard disk.

You can use YaST (as root) to perform this task by doing the following:

a. Start the Expert Partitioner from the YaST Control Center (**System > Partitioner**) or from the command line (**yast2 disk**).

b. At the warning message, select **Yes**.

   The Expert Partitioner dialog appears.

c. Create a new partition by selecting **Create**; then select **Primary partition** and select **OK**.

d. Select **Do not format**; then for the File system ID, select **0x8E Linux LVM**.

e. Accept the default setting in the **Start cylinder** field; then enter a size value in the **End** field.

f. When you finish, select **OK**.

   The LVM partition appears in the partition list.

g. Configure the partition by selecting **Apply**; then select **Finish**.

2. Create the volume group and logical volumes:

a. Start the LVM module from the YaST Control Center (**System > LVM**) or from the command line (**yast2 lvm_config**).

The following appears:

Figure 3-12

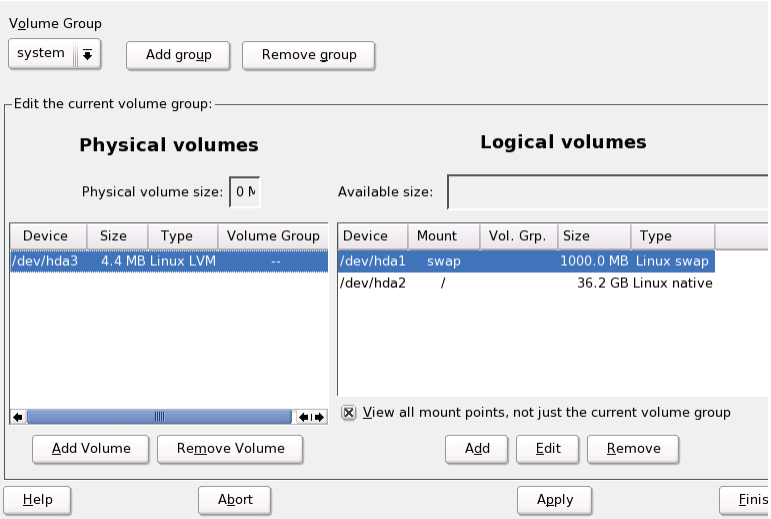
The **Volume Group Name** is the name you want to use for selecting and managing the volume group.

The **Physical Extent Size** value is normally set to 4 MB, which allows for a maximum size of 256 GB for physical and logical volumes. You should only increase this value if you need logical volumes larger than 256 GB.

b. Accept the default ***volume group name*** and ***physical extent size***, or enter your own settings; then select **OK**.

The following appears:

Figure 3-13



c. From the **Volume Group** drop-down list, select the ***volume group*** you want to configure.

You can add other volume groups (**Add group**) or delete a volume group (**Remove group**), but you can delete a volume group only if it has no partitions assigned.

Only 1 volume group needs to be created for a normally installed SUSE Linux system.

The Physical volumes list includes all partitions with either the Linux LVM or Linux native type (no swap or DOS partitions appear).

If a partition is already assigned to a volume group, the name of the volume group is shown in the list. Unassigned partitions are indicated by “--”.

d. Add an unassigned partition to the volume group by selecting the ***partition***; then select **Add Volume**.

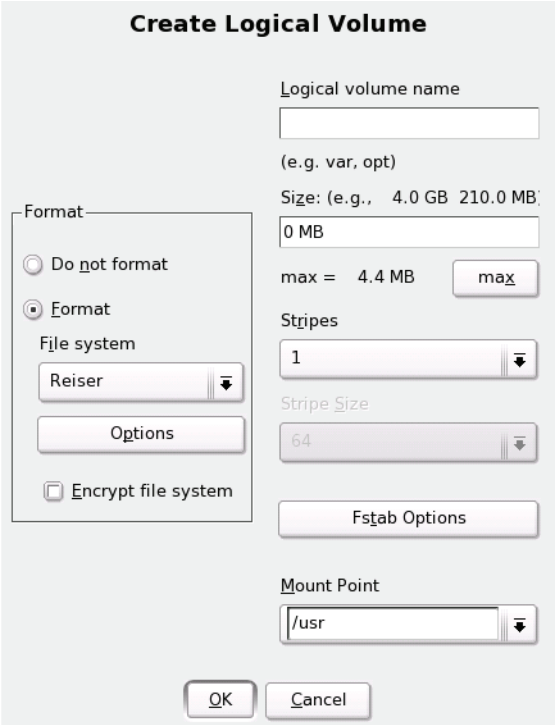
The name of the volume group is entered next to the selected partition.

The Logical volumes list includes all logical volumes defined in the volume group. In addition, you can view all mount points (including traditional file systems) by selecting **View all mount points**.

e. Create a logical volume by selecting **Create**.

The following appears:

Figure 3-14



This dialog lets you configure a logical volume using the same options available for creating a file system (see "Create a File System From YaST" on 3-58).

In addition, you can enter a logical volume name, the maximum amount of space available (by selecting **max**), the number of stripes, and the stripe size (if you configure more than one stripe).

You can create a logical volume with multiple stripes only if the hard disk space required by the logical volume can be distributed evenly to multiple physical volumes.

If only 2 physical volumes are available, a logical volume with 3 stripes is impossible.

---

f. When you finish configuring the logical volume, select **OK**.

   You are returned to the LVM configuration dialog.

   The new logical volume appears in the logical volumes list.

g. Complete the configuration by selecting **Finish**.

   A dialog appears indicating that all settings have been written and the volume is ready to use.

h. Select **OK**.

For additional information on configuring LVM, see the LVM HOWTO at http://tldp.org/HOWTO/LVM-HOWTO/.

Exercise 3-2 Create Logical Volumes

In this exercise, you use YaST to create 2 physical volumes (PV) to add to a volume group (VG) named project1. Within the volume group, you add 2 logical volumes named pilot (750MB) and prod (750MB).

Do the following:

- Part I: Create LVM Volumes
- Part II: Resize an LVM Volume

Part I: Create LVM Volumes

Do the following:

1. From the KDE desktop, select the **YaST** icon; then enter a password of **novell** and select **OK**.

   The YaST Control Center appears.

2. Select **System > Partitioner**.

   A warning message appears.

3. Continue by selecting **Yes**.

   The Expert Partitioner dialog appears.

4. Create a new LVM partition by doing the following:

   a. Select **Create**.

   b. Select **Do not format**; then select or enter the following

      - File system ID: **0x8E Linux LVM**
      - End (cylinder): **+1G**

   c. Save the partition definition by selecting **OK**.

5. Create another 1GB LVM partition by repeating step 4.

   You should now have 2 1GB LVM partitions.

6. Write the changes to the partition table by selecting **Apply**; then select **Finish**.

   The partitioning is configured.

7. From the YaST Control Center, start the YaST LVM module by selecting **System > LVM**.

   If you start the LVM module through the Expert Partitioner, the following steps will not work properly.

   A Create a Volume Group dialog appears.

8. Enter the following:
   - Volume Group Name: **project1**
   - Physical Extent Size: **4M**

9. Continue by selecting **OK**.

   A dialog with lists of physical volumes and logical volumes appears. The LVM partitions you created are listed under physical volumes Linux LVM in the Type column.

10. Add each Linux LVM physical volume to the volume group project1 by selecting each physical volume (such as **/dev/hda8**) and then selecting **Add Volume**.

    Scroll to the right (if necessary) and notice that both physical volumes are listed with a volume group of project1.

11. Add a logical volume pilot from the project1 volume group:

    a. Select **Add**.

       A Create Logical Volume dialog appears.

b. Enter or select the following:
   - Format (File system): **Reiser**
   - Logical volume name: **pilot**
   - Size: **750 MB**
   - Mount Point: **/project1/pilot**

c. Save the logical volume definition by selecting **OK**.

12. Add a logical volume prod from the project1 volume group:

a. Select **Add**.

A Create Logical Volume dialog appears.

b. Enter or select the following:
   - Format (File system): **Reiser**
   - Logical volume name: **prod**
   - Size: **750 MB**
   - Mount Point: **/project1/prod**

c. Save the logical volume definition by selecting **OK**.

13. Save the changes by selecting **Finish**.

A message appears, indicating that the process is complete.

14. Continue by selecting **OK**.

15. From a terminal window, su to root (**su -**) with a password of **novell**.

16. View the new LVM file systems by entering the following:

df -h

Notice the size of these new file systems.

17. View the device names and mount locations by entering **cat /etc/fstab**.

18. Change to the directory /project1/prod/ by entering **cd /project1/prod**.

19. Copy all binary files from /usr/bin to this directory by entering the following (make sure you include the period):

    cp -v /usr/bin/* .

20. View the space used and available for prod by entering **df -h**.

Part II: Resize an LVM Volume

You now require more disk space on the prod LVM file system. In the following steps you allocate the remaining space to prod.

Doing the following:

1. From the YaST Control Center, select **System > LVM**.

   The LVM dialog appears.

2. From the Logical volumes list select **/dev/project1/prod**; then select **Edit**.

   The Edit Logical Volume dialog appears.

   Notice the volume size (752.0 MB).

3. Select the **max** button.

   Notice that the size changes to the maximum space available (1.2 GB).

4. Continue by selecting **OK**.

5. Save the changes by selecting **Finish**; then select **OK**.

6. From the terminal window, view the new size of prod by entering **df -h**.

7. Close all open windows.

(End of Exercise)

Objective 4 Configure and Manage a Linux File System

To perform basic Linux file system management tasks in SUSE Linux Enterprise Server, you need to know how to do the following:

- Create a File System From YaST
- Create a File System From the Command Line
- Mount a File System
- Monitor and Check a File System
- Create a Boot, Rescue, or Module Disk

Create a File System From YaST

After creating a partition (during or after installation), you can use YaST to assign the partition a file system (such as ext3 or ReiserFS) by doing the following:

1. From the KDE desktop, start the YaST Expert Partitioner module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > Partitioner**.

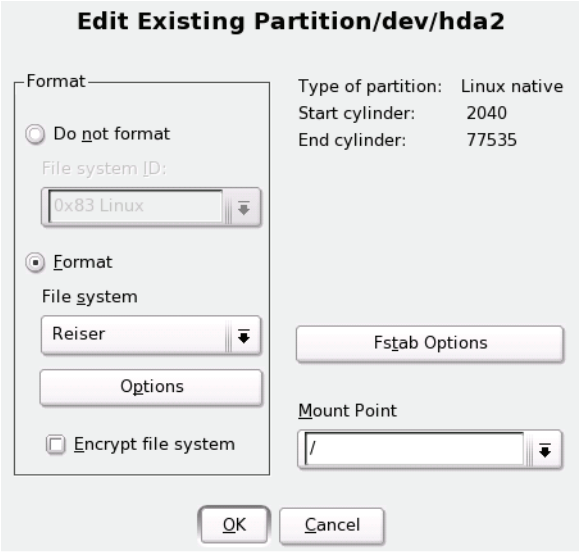
     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 disk**.

   The Expert Partitioner appears.
2. Select the ***partition*** you want to assign a file system; then select **Edit**.

   The Edit dialog for the partition appears.
3. Format the partition with a file system by selecting **Format**.

The following appears:

Figure 3-15



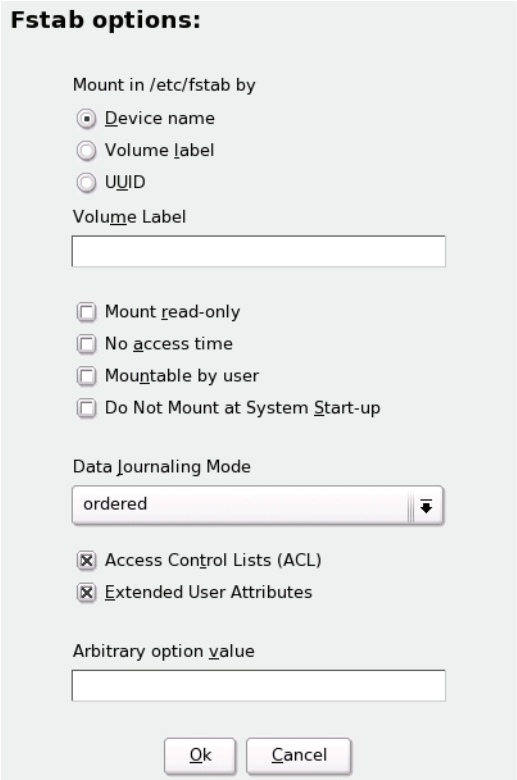
4. From the **File system** drop-down list, select an available file system (such as **Reiser** or **Ext3**).
5. View the available format options by selecting **Options**; then return to the main format menu by selecting **OK**.

These options include the hash function to use (for sorting filenames in directories) and the ReiserFS format revision to use. We recommend keeping the default settings for most implementations

6. Encrypt all data saved to the partition by selecting **Encrypt file system**.
7. Select file system options to configure in the /etc/fstab file by selecting **Fstab Options**.

The following appears:

Figure 3-16



These options are saved in /etc/fstab and are used when mounting the file system.

A description of each option is included in the left frame of the Fstab options dialog.

8. When you finish configuring the fstab options; select **Ok**.
9. In the **Mount Point** field enter the ***directory*** where the partition should be mounted in the file system tree.

10. When you finish configuring the file system and mounting parameters, select **OK**.

    A warning message appears cautioning you about committing the changes you've made.

11. Select **Continue**.

12. Save the changes by selecting **Apply**.

    Another warning message appears asking if you really want to make the changes.

13. Configure the partition with the new file system and make the changes to the fstab file by selecting **Finish**.

Create a File System From the Command Line

You can use the following commands to create a file system from the command line:

- mkfs
- mkreiserfs

mkfs

You can create a file systems (such as ext2, ext3, MS-DOS, MINIX, XFS, JFS) with the command mkfs (make file system).

This command is a frontend for the commands you use to create file systems (such as mkfs.ext2, mkfs.ext3, or mkfs.msdos). For this reason, you need to use the option -t to indicate the file system type you want to create.

If you do not indicate a file system type, mkfs automatically creates an ext2 file system.

If you create an ext2 or ext3 file system with mkdir, you can use the following options:

Table 3-7

Option	Description
-b *blocksize*	You can use this option to indicate the size of the data blocks in the file system. Values of 1024, 2048, . . . , 16384 are allowed for the block size.
-i *bytes_per_inode*	You can use this option to indicate how many inodes are created on the file system. For *bytes_per_inode* you can use the same values available for the block size. You should choose a larger value for the block size. However, it makes little sense to have a larger number of inodes than data blocks.
-j	You can use this option to create an ext3 Journal on the file system.

If you do not include options -b and -i, the data block sizes and the number of inodes is set by mkfs, depending on the size of the partitions.

The following is an example of creating a partition with an ext2 file system:

```
earth:~ # mkfs -t ext2 /dev/hdb1
mke2fs 1.34 (25-Jul-2003)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
25688 inodes, 102400 blocks
5120 blocks (5.00%) reserved for the super user
First data block=1
13 block groups
8192 blocks per group, 8192 fragments per group
1976 inodes per group
Superblock backups stored on blocks:
8193, 16385, 24577, 32769, 40961, 49153, 57345, 65537,
73729, 81921, 90113, 98305
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 31 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
earth:~ #
```

This mkfs example creates a 100 MB partition formatted with the following standard values:

- **Block size=1024 (log=0)**

  The block size is 1 KB.

- **25688 inodes, 102400 blocks**

  The maximum number of files and directories is 25688. The total number of blocks is 102400.

- **5120 blocks (5.00%) reserved for the super user**

  5% of the entire space is reserved for the system administrator. If the hard disk is 95% full, then a normal user cannot use any more space.

You can also use the command mke2fs (which corresponds to mkfs.ext2 and mkfs.ext3) to create an ext2 or ext3 file system (see **man mke2fs**).

mkreiserfs

You can create a Reiser file system by using the command mkreiserfs.

The following are a commonly-used parameter and option for mkreiserfs:

Table 3-8

Option	Description
number_of_blocks	This parameter represents the size of the partition in number of blocks. If you do not include this parameter, mkreiserfs automatically sets the block size.
--format *format*	You can use this option to specify the format of the Reiser file system (**3.5** or **3.6**).

Mount a File System

Instead of using separate drive letters to represent different partitions in the file system (such as MS-DOS and Windows® 9*x*), Linux mounts partitions in a folder in the file system using *mount points*.

For example, to add a new hard disk to a Linux system, first you would partition and format the drive. You would then create a directory (such as /mnt/files/) in the file system and mount the drive in that directory using the command mount.

To unmount (detach) a file system, you use the umount command (for details, enter **man umount**).

You can also mount remote file systems, shared via the Network File System (NFS), to directories you create in your file system.

The directory /mnt/ is used by default for mounting local and remote file systems. All removable devices are mounted by default to /media/, such as the following:

- A CD-ROM on /dev/cdrom is mounted by default to /media/cdrom.
- A floppy disk on /dev/floppy is mounted by default to /media/floppy.

When using SLES 9 from a desktop environment such as KDE, media such as floppy disks and CDs are automatically mounted and unmounted using the defaults in /etc/fstab and the feature submount/subfs.

The file systems that automatically mount and unmount contain a subfs parameter in the line, as in the following:

Figure 3-17

```
/dev/hda2          /                  reiserfs   acl,user_xattr
/dev/hda1          swap               swap       pri=42
devpts             /dev/pts           devpts     mode=0620,gid=5
proc               /proc              proc       defaults
usbfs              /proc/bus/usb      usbfs      noauto
sysfs              /sys               sysfs      noauto
/dev/cdrom         /media/cdrom       subfs      fs=cdfss,ro,procuid,n
/dev/fd0           /media/floppy      subfs      fs=floppyfss,procuid,
```

After the media is mounted to the directory in the file system, you can access the content on the media by changing to that directory.

To understand how to manage mounting (and unmounting) file systems, you need to know the following:

- Configuration Files for Mounting
- How to View Currently-Mounted File Systems
- How to Mount a File System

- How to Mount a File System in More Than One Location
- How to Unmount a File System

Configuration Files for Mounting

The file systems and their mount points in the directory tree are configured in the file /etc/fstab. This file contains 1 line with 6 fields for each mounted file system.

The lines look similar to the following:

```
Field 1    Field 2       Field 3  Field 4         Field 5 Field 6

/dev/hda2  /             reiserfs acl,user_xattr  1       1
/dev/hda1  swap          swap     pri=42          0       0
devpts     /dev/pts      devpts   mode=0620,gid=5 0       0
proc       /proc         proc     defaults        0       0
usbfs      /proc/bus/usb usbfs    noauto          0       0
sysfs      /sys          sysfs    noauto          0       0
/dev/cdrom /media/cdrom   subfs    fs=cdfss,ro,procuid,nosuid,nodev,exec,
/dev/fd0   /media/floppy  subfs    fs=floppyfss,procuid,nodev,nosuid,sync
```

Each field provides the following information for mounting the file system:

- **Field 1.** The name of the device file.
- **Field 2.** The mount point—the directory to which the file system should be mounted. The directory specified here must already exist.
- **Field 3.** The file system type (such as ext2 or reiserfs).
- **Field 4.** The mount options. Multiple mount options are separated by commas (such as **fs=cdfss,ro,procuid,nosuid**).

  For example, the options **ro** and **nodev** for the CDROM drive (/dev/cdrom) mean that the drive is read only (**ro**) and that device files on the CD are not interpreted as such by the file system (**nodev**).

- **Field 5.** Indicates whether to use the backup utility dump for the file system. **0** means no backup.
- **Field 6.** Indicates the sequence of the file system checks (with the **fsck** utility) when the system is booted:
  - **0:** file systems that are not to be checked
  - **1:** the root directory
  - **2:** all other modifiable file systems

While /etc/fstab lists the file systems and their mount points in the directory tree, the /etc/mstab file lists the file systems currently mounted and their mountpoints.

The mount and umount commands modify the /etc/mtab file and affect the state of mounted file systems.

The kernel also keeps information for /proc/mounts, which lists all currently mounted partitions.

For troubleshooting purposes, if there is a conflict between /proc/mounts and /etc/mstab information, the /proc/mount data is always more current and reliable than /etc/mstab.

How to View Currently-Mounted File Systems

You can view the file systems currently mounted in SUSE Linux by entering the command mount. Information similar to the following appears:

```
DA50:~ # mount
/dev/hda2 on / type reiserfs (rw,acl,user_xattr)
proc on /proc type proc (rw)
tmpfs on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hdc on /media/cdrom type subfs (ro,nosuid,nodev,fs=cdfss,procuid,
iocharset=utf8)
/dev/fd0 on /media/floppy type subfs (rw,nosuid,nodev,sync,fs=floppyfss,
procuid)
usbfs on /proc/bus/usb type usbfs (rw)
```

You can also view this information in the file /proc/mounts.

How to Mount a File System

You can use the command mount to manually mount a file system. The general syntax for mounting a file system with mount is

mount [-t *file_system_type*] [-o *mount_options*] *device mount_point_directory*

By using mount, you can override the default settings in /etc/fstab.

For example, entering the following mounts the partition /dev/hda9 on the directory /space/:

```
mount /dev/hda9 /space
```

You do not usually specify the file system type because it is recognized automatically (using magic numbers in the superblock):

The following are some of the options you can use when mounting a file system with the command mount or by entering them in /etc/fstab:

- **remount.** This option causes file systems that are already mounted to be mounted again.

  When you make a change to the options in /etc/fstab, you can use remount instead of rebooting the system to incorporate the changes.

- **rw, ro.** These options indicate whether a file system should be writable (**rw**) or only readable (**ro**).

- **sync, async.** These options set synchronous (**sync**) or asynchronous (**async**) input and output in a file system. The default setting is async.

- **atime, noatime.** These options set whether the access time of a file is updated in the inode (**atime**) or not (**noatime**). The option noatime should improve the performance.

- **nodev, dev.** The **nodev** option prevents device files from being interpreted as such in the file system.

- **noexec, exec.** You can prohibit the execution of programs on a file system with the option **noexec**.

- **nosuid, suid.** The **nosuid** option ensures that the suid and sgid bits in the file system are ignored.

Some options only make sense in the file /etc/fstab. These options include the following:

- **auto, noauto.** File systems set with the option **noauto** in the file /etc/fstab are not mounted automatically when the system is booted. These are usually floppy disk drives or CD-ROM drives.

- **user, nouser.** This option lets users mount the file system. Normally, this is a privilege of the user root.

- **defaults.** This option causes the default options rw, suid, dev, exec, auto, nouser, and async to be used.

The options **noauto** and **user** are usually combined for exchangeable media such as floppy disk or CD-ROM drives.

How to Mount a File System in More Than One Location

Once you mount a file system, you can remount it to a new location in the file system using the --bind parameter (since Linux 2.4.0).

The syntax for using --bind is

mount --bind *old_directory new_directory*

One way to make sure the file system is not busy is to enter **cd /** at the shell prompt before using the umount command. This command takes you to the root of the file system.

For example to mount the /home/bwayne directory to /tmp/home, you would enter the following:

mount --bind /home/bwayne /tmp/home

The mount --bind command is also useful for protecting services such as DNS by mounting them in the /chroot directory.

To help determine the process that are acting on a file or directory, you can use the fuser utility. For details on using the fuser utility, see "Check PID Usage (fuser)" on 3-79.

How to Unmount a File System

Once a file system is mounted, you can use the **umount** command (without an "n") to unmount the file system.

You can unmount the file system by using umount with the device or the mount point.

For example to unmount a CD file system mounted at /media/cdrecorder, you could enter one of the following:

- **umount /media/cdrecorder**

  or

- **umount /dev/cdrecorder**

In order to unmount the file system, you should not be currently at the mountpoint in the shell prompt. If you are at the mountpoint, Linux sees the file system as being "busy" and will refuse to unmount the file system.

One way to make sure the file system is not busy is to enter **cd /** at the shell prompt before using the umount command. This command takes you to the root of the file system.

However, there might be times when the system (kernel) still sees the file system as busy, no matter what you try to do.

In these cases, you can enter umount -f to force the file system to unmount. However, we recommend using this as a last resort, as there is probably a reason why the kernel thinks the file system is still mounted.

To help determine the process that are acting on a file or directory, you can use the fuser utility. For details on using the fuser utility, see "Check PID Usage (fuser)" on 3-79.

Exercise 3-3 Manage File Systems From the Command Line

In this exercise, you manage file systems from the command line by doing the following:

- Part I: Create File Systems
- Part II: Customize the File Systems

Part I: Create File Systems

Do the following:

1. From the KDE desktop, open a terminal window; then su to root (**su -**) with a password of **novell**.
2. Create the following file systems:
   a. Create a new FAT32 file system on /dev/hda5 by entering the following:

      mkfs.msdos -n data1 /dev/hda5

      A message such as mkfs.msdos 2.10 (22 Sep 2003) confirms the file system creation.

   b. Create a new EXT2 file system on /dev/hda6 by entering the following:

      mkfs -t ext2 -v /dev/hda6

      Notice that by adding the option -v, you received extensive information about the new file system.

   c. Create a new Reiser file system on /dev/hda7 that is only 625 MB by entering the following:

      mkreiserfs /dev/hda7 160000

      A warning message appears indicating that all data will be lost on /dev/hda7.

   d. Continue by entering **y**.

3. Add entries to the file /etc/fstab for the new file systems:
   a. Make sure a directory named /export/ exists by entering **ls -ld /export**.
   b. Change to the directory /export/ by entering **cd /export**.
   c. Create the directories **data1** and **data3** under /export/ by using **mkdir**.
   d. Verify that the directories were created by entering **ls -l**.
   e. Open the file /etc/fstab in an editor by pressing **Alt+F2**, entering **kdesu kate /etc/fstab**, and selecting **Run**; then enter a password of **novell** and select **OK**.
   f. At the end of the file fstab, add the following entries:

      /dev/hda5 /export/data1 vfat defaults 1 2
      /dev/hda6 /export/data2 ext2 defaults 1 2
      /dev/hda7 /export/data3 reiserfs defaults 1 2

      You must include an empty line at the end of the file, otherwise the mount command cannot read the file.

      These entries make sure that the hda5, hda6, and hda7 partitions are mounted when starting or rebooting the system.

      The hda4 partition is the extended partition and does not need to be mounted.
   g. When you finish, select **File > Save** (keep /etc/fstab open).
4. From the terminal window, mount all of the new file systems and re-read the /etc/fstab file by entering **mount -a**.
5. View the information about the mounted file systems by entering the following 3 commands:

   mount
   df -h
   cat /etc/mtab

Part II: Customize the File Systems

In this part of the exercise, you convert the partition /dev/hda6 to an ext3 file system by adding a journal. You also resize the /dev/hda7 Reiser file system to consume the entire partition and not just 625 MB.

Do the following:

1. Modify the partition /dev/hda6:
   a. From the terminal window, view details about the ext2 file system on /dev/hda6 by entering the following:

      dumpe2fs /dev/hda6 | more

      Notice the block size and the file system state.
   b. Give the ext2 file system the volume name /export/data2 while the file system is mounted by entering the following:

      tune2fs -L /export/data2 /dev/hda6

      Naming a file system can be useful in system rescue situations when the /etc/fstab is not available. It is common practice to use this naming convention.
   c. Verify that the file system now has a volume name by entering **dumpe2fs /dev/hda6 | less**.
   d. Add a journal to the file system (making it an ext3 file system) by entering **tune2fs -j /dev/hda6**.
   e. Verify that the file system now contains a journal by entering **dumpe2fs /dev/hda6 | less**.
   f. View information about the mounted file systems by entering **mount**.

      Notice that the file system is still mounted as an ext2 file system.
   g. Unmount the partition /dev/hda6 by entering **umount /dev/hda6**.
   h. Verify that the file system state is clean by entering **dumpe2fs /dev/hda6 | less**.

i. From the Kate window, edit the file /etc/fstab to change the file system type from ext2 to ext3, as in the following:

   /dev/hda6 /export/data2 **ext3** defaults 1 2

j. Select **File > Save**.

k. From the command line, re-read /etc/fstab and mount the partition as an ext3 file system by entering **mount -a**.

l. Verify the change by entering **mount**.

m. Unmount the partition /dev/hda6 again by entering **umount /export/data2**.

n. Mount the partition as an ext2 file system manually by entering the following:

   mount -t ext2 /dev/hda6 /export/data2

o. Verify that the file system is mounted without a journal (as an ext2 file system) by entering **mount**.

p. Remount /dev/hda6 as an ext3 file system and verify the change by entering the following 3 commands:

   umount /export/data2
   mount -a
   mount

2. Modify the partition /dev/hda7:

a. View the size of the partition /dev/hda7 by entering **df -h**.

b. Unmount dev/hda7 so that the Reiser file system on it can be resized to fill the entire partition by entering **umount /export/data3**.

c. While the partition is unmounted, add a label to the file system of /export/ data3 by entering the following:

   reiserfstune -l /export/data3 /dev/hda7

d. Resize the partition to consume the entire partition by entering **resize_reiserfs /dev/hda7**.

e. Remount the partition by entering **mount -a**.

f. View the size of the partition by entering **df -h**.

The size is no longer 625 MB, but should be 1 GB or more depending on the size of your hard drive.

g. Unmount the partition to run a file system check on it by entering **umount /export/data3**.

h. Run a check on the file system on /dev/hda7 by entering the following:

reiserfsck -y /dev/hda7

i. Remount all file systems by entering **mount -a**.

3. Close all open windows.

(End of Exercise)

Monitor and Check a File System

Once you set up and begin using your Linux file system, you can monitor the status and health of the system by doing the following from the command line:

- Check Partition and File Usage (df and du)
- Check Open Files (lsof)
- Check PID Usage (fuser)
- Check /lost+found (ext2 and ext3 only)
- Check and Repair Any File System (fsck)
- Check and Repair ext2/ext3 and ReiserFS (e2fsck and reiserfsck)

Check Partition and File Usage (df and du)

The following commands help you monitor usage by partitions, files, and directories:

- **df.** This command provides information on where hard drives and their partitions or other drives are mounted in the file system, and how much space they occupy.

  If you do not include a filename, the space available on all currently-mounted file systems is displayed.

  If you provide the filename of a disk device node containing a mounted file system, df displays the space available on that file system rather than on the file system containing the device node (which is always the root file system).

  Some useful options include **-h** (human readable format), **-i** (list inode information instead of block usage), and -l (limit listing to local file systems).

  For example, to list information for all local file systems in human-readable format, you would enter **df -lh**.

- **du.** This command provides information on the space occupied by files and directories.

  Some useful options include **-c** (display a grand total), **-h** (human-readable format), **-s** (display only a total for each argument), and **--exclude=*pattern*** (exclude files that match ***pattern***).

  For example to display information for files in human-readable format except for files that end in ".o," you would enter the following:

  du -h --exclude='*.o'

Check Open Files (lsof)

The command lsof lists open files. Entering **lsof** without any options, lists all open files belonging to all active processes.

An open file can be a regular file, a directory, a block special file, a character special file, a library, and a stream or a network file (Internet socket, NFS file, or UNIX domain socket.)

In addition to producing a single output list, lsof can run in repeat mode. In repeat mode it outputs, delays, and then repeats the output operation until stopped with an interrupt or quit signal.

Some useful options include **-c *x*** (list only files starting with ***x***), **-s** (display file sizes), and **-u *x*** (list only files for users who are ***x***).

For example to list open files for the users root and geeko only and include the file sizes, you would enter **lsof -s -u root,geeko**.

Check PID Usage (fuser)

The command fuser displays the PIDs of processes using the specified files or file systems.

In the default display mode, each filename is followed by a letter that describes the type of access:

- **c:** current directory
- **e:** executable being run
- **f:** open file (omitted in default display mode)
- **r:** root directory
- **m:** mmaped file or shared library

A non-zero return code is displayed if none of the specified files is accessed or in case of a fatal error. If at least one access has been found, fuser returns zero.

Some useful options include **-a** (return information for all files, even if it they are not accessed by a process), **-v** (verbose mode), and **-u** (append the user name of the process owner to each PID).

For example, to check the PID information for hosts, even if no process is currently associated with the file, you would enter **fuser -a /etc/hosts**.

Check /lost+found (ext2 and ext3 only)

The directory /lost+found/ is a special feature of the ext2 and ext3 file system format. This directory is used to save files or file fragments that have “gone missing” (such as after a system crash).

After a system crash, Linux automatically carries out a check of the complete file system. Files or file fragments to which a name can no longer be allocated are not simply deleted, but stored in this directory.

By reviewing the contents of this directory, you can try to reconstruct the original name and purpose of a file.

Check and Repair Any File System (fsck)

The command fsck lets you check and optionally repair one or more Linux file systems. Normally, fsck tries to run file systems on different physical disk drives in parallel to reduce total amount time to check all of the file systems.

If you do not specify a file system on the command line, and the -A option is not included, fsck defaults to checking filesystems in /etc/fstab.

After checking, an exit code is returned that lists 1 or more of the following conditions:

Table 3-9

Exit Code	Description
0	No errors
1	File system errors corrected
2	System should be rebooted
4	File system errors left uncorrected
8	Operational error
16	Usage or syntax error
32	Fsck canceled by user request
128	Shared library error

In reality, fsck is simply a frontend for the various file system checkers (**fsck.*fstype***) available the system. The fsck utility looks for the system-specific checker in /sbin/ first, then in /etc/fs/ and /etc/, and finally in the directories listed in the PATH environment variable.

To check a specific file system, using the following syntax:

fsck *filesystem*

For example if you wanted to check the file system /dev/hda2, you would enter **fsck /dev/hda2**.

Some options that are available with fsck include -A (walk through the /etc/fstab file and try to check all the file systems in one pass), -N (don't execute, just show what would be done), and -V (verbose output).

Check and Repair ext2/ext3 and ReiserFS (e2fsck and reiserfsck)

Switching off the Linux system without unmounting partitions (for example, when a power outage occurs) can lead to errors in the file system.

The next time you boot the system, the fact that the computer was not shut down correctly is detected and a file system check is performed.

If errors are found in the file system, the rescue system may need to be used. Depending on the file system type, you use either /sbin/e2fsck or /sbin/reiserfsck.

These tools check the file system for a correct superblock (the block at the beginning of the partition containing information on the structure of the file system), faulty data blocks, or faulty allocation of data blocks.

A possible problem in the ext2 (or ext3) file system is damage to the superblock. You can first view the location of all copies of the superblock in the file system using dumpe2fs.

Then, with e2fsck, you can copy one of the backup copies to the beginning of the file system, as in the following:

```
e2fsck -f -b 32768 /dev/hda1
```

In this example, the superblock located at data block 32768 in the ext2 file system of the partition/dev/hda1 is copied to the beginning of the file system.

Normally a backup copy of the superblock is stored every 32768 blocks.

With reiserfsck, the file system is subjected to a consistency check. The journal is checked to see if certain transactions need to be repeated. With the option --fix-fixable, errors such as wrong file sizes are fixed as soon as the file system is checked.

With an error in the binary tree, it is possible to have this rebuilt by entering **reiserfsck --rebuild-tree**.

Create a Boot, Rescue, or Module Disk

In case of a system failure (such as a corrupted file system), you normally insert the CD 1 of the installation media and select rescue system.

CD2 also has a boot capability that works if CD1 is not available.

However, if there is no bootable CD-ROM drive, you can create a set of floppy disks to boot the machine and start a rescue system.

To create a boot disk, rescue disk, or module disk with YaST, do the following:

1. From the KDE desktop, start the YaST Create a Boot, Rescue, or Module Floppy module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > Create a Boot, Rescue, or Module Floppy**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 bootfloppy**.

   The following appears:

Figure 3-18

These floppy disks are helpful if the boot configuration of your system is damaged. The rescue disk is especially necessary if the file system of the root partition is damaged.

In this case, you might also need the module disk with various drivers to be able to access the system (such as accessing a RAID system).

2. Select one of the following:

   - **Standard Boot Disk.** Select this option to create a standard boot disk with which to boot an installed system. This disk is also needed for starting the rescue system.
   - **Rescue Disk.** This disk contains a special environment that allows you to perform maintenance tasks in your installed system, such as checking and repairing the file system and updating the boot loader.

     To start the rescue system, boot with the standard boot disk then select **Manual Installation > Start Installation or System > Rescue System**.

     You are then prompted to insert the rescue disk. If your system was configured to use special drivers (such as RAID or USB), you might need to load the respective modules from a module disk.
   - **Module Disks.** Module disks contain additional system drivers. The standard kernel only supports IDE drives.

     If the drives in your system are connected to special controllers (such as SCSI), load the needed drivers from a module disk.

     If you select this option and select **Next**, a dialog is displayed for creating various module disks.

     The following module disks are available:

     - **USB Modules.** This floppy disk contains the USB modules you might need if USB drives are connected.

- **IDE, RAID, and SCSI Modules.** As the standard kernel only supports normal IDE drives, you need this module disk if you use special IDE controllers. In addition, all RAID and SCSI modules are provided on this disk.
- **Network Modules.** If you need access to a network, load the suitable driver module for your network card from this floppy disk.
- **PCMCIA, CD-ROM (non-ATAPI), FireWire, and File Systems.** This floppy disk contains all PCMCIA modules used especially for laptop computers. In addition, the modules for FireWire and some less common file systems are available here. Older CD-ROM drives that do not comply with the ATAPI standard can also be operated with drivers from this floppy disk.

To load drivers from a module disk to the rescue system, select **Kernel Modules (hardware drivers)** and the desired module category (such as SCSI or ethernet). You are prompted to insert the respective module disk and the contained modules are then listed. Select the desired module.

Watch the system messages carefully. For example, the message Loading module <modulename> failed indicates that the hardware could not be recognized by the module.

Some older drivers require specific parameters to be able to address the hardware correctly. In this case, refer to the documentation for your hardware.

- **User-Defined Disk.** Select this option to write any existing floppy disk image from the hard disk to a floppy disk.
- **Download Disk Image.** Select this option to enter a URL and authentication data to download a floppy disk image from the Internet.

3. After selecting a floppy disk option, continue by selecting **Next**.
4. When prompted, insert the floppy disk.
5. Create the floppy disk by selecting **Next**.

Objective 5 Set Up and Configure Disk Quotas

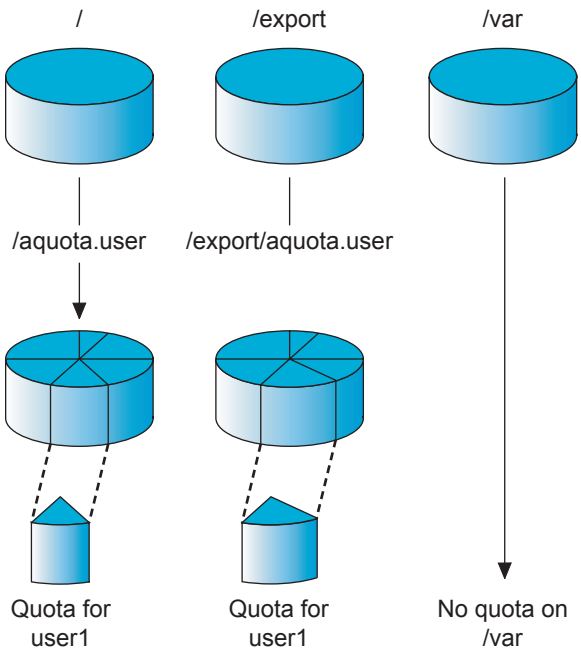
Drive space continues to be a problem, especially when storing data such as user files, databases, and MP3 archives. Without imposing limits, a user can easily fill up a 40 GB of hard drive space with pictures, software, and music.

Linux includes a quota system that lets you specify a specific amount of storage space for each user or group, and how many files that user or group can create.

In SUSE Linux Enterprise Server you can use the quota package to establish these limitations.

The following illustrates the quota architecture:

Figure 3-19



Disk quota support is already included in the kernel in SUSE Linux Enterprise Server. You can implement disk quotas for partitions configured with the ext2, ext3, or ReiserFS file systems.

Setting up and configuring the disk quota service on your server includes the following tasks (in order):

- Prepare the File System
- Initialize the Quota System
- Configure and Manage User and Group Quotas
- Start and Activate the Quota Service

Prepare the File System

When the system is started, the quotas for the file system must be activated. You can indicate for which file system's quotas are to be activated by configuring entries in the file /etc/fstab.

You enter the keyword **usrquota** for quotas on the user level and the keyword **grpquota** for group quotas, as in the following:

Figure 3-20

```
/dev/hda2          /                  reiserfs   acl,user_xattr,usrquota,grpquota
/dev/hda1          swap               swap       pri=42                  0 0
devpts             /dev/pts           devpts     mode=0620,gid=5         0 0
proc               /proc              proc       defaults                0 0
usbfs              /proc/bus/usb      usbfs      noauto                  0 0
sysfs              /sys               sysfs      noauto                  0 0
/dev/cdrom         /media/cdrom       subfs      fs=cdfss,ro,procuid,nosuid,nodev,ex
/dev/fd0           /media/floppy      subfs      fs=floppyfss,procuid,nodev,nosuid,s
```

In this example, quotas are configured for the file system / (root).

If you have configured /etc/fstab without rebooting your server, you need to remount the file systems in the root partition by entering the following:

```
mount -o remount /
```

Initialize the Quota System

After remounting, you need to initialize the quota system. You can do this by using the command **quotacheck**, which is part of the package quota.

This command checks the partitions with quota keywords (in terms of already occupied data blocks and inodes) and stores the determined values in the files **aquota.user** (for user quotas) and **aquota.group** (for group quotas).

Up to kernel version 2.4 these files were called quota.user and quota.group, and had to be created before quotacheck was run.

If you enter the command **quotacheck -avug**, all mounted file systems (**-a**) are checked for data blocks and inodes that are occupied by users (**-u**) and groups (**-g**). The option **-v** provides a detailed output.

When checking mounted file systems, you might need to use the option **-m** to force the check.

Assuming the quota entries exist for / and /export, after running **quotacheck** the following files are created:

```
DA50:~ # ls -l /aquota*
-rw-------  1 root root 9216 Aug 27 10:06 /aquota.group
-rw-------  1 root root 9216 Aug 27 10:06 /aquota.user
```

Configure and Manage User and Group Quotas

To configure quotas for users and groups, you need to know how to do the following:

- Configure Soft and Hard Limits for Blocks and Inodes
- Configure Grace Periods for Blocks and Inodes

- Copy User Quotas
- Generate a Quota Report

Configure Soft and Hard Limits for Blocks and Inodes

With the command edquota and the following options, you can edit the current quota settings for a user or group:

- **edquota -u *user***: for setting up user quotas.
- **edquota -g *group***: for setting up group quotas.

The current settings are displayed in the vi editor for you to edit. You can edit the soft and hard limits; however, the blocks and inodes values are for information only and cannot be edited.

For example, you can enter the following to configure quotas for the user **geeko**:

edquota -u geeko

After entering the command, the following quota information appears in vi for geeko:

```
Disk quotas for user geeko (uid 1001):
  Filesystem       blocks    soft    hard    inodes    soft    hard
  /dev/sda2          7820   10000   20000       145       0       0
```

The following describes the settings:

- **Blocks.** How much hard disk space is currently used, with soft and hard limits listed.

  The values for blocks are given in blocks of 1 KB (independent of the block size for the ext2 file system).

  For example, the value **7820** under Blocks indicates that the user geeko is currently using about 8 MB of hard drive space.

Notice that the soft limit is set to **10** MB and the hard limit is set to **20** MB.

- **Inodes.** How many files belong to the user on the file system, with soft and hard limits listed.

  Notice that the soft and hard limits for geeko are set to **0**, which means that the user can create an unlimited number of files.

The soft limits indicate a quota that the user cannot permanently exceed. The hard limits indicate a boundary beyond which no more space or inodes can be used.

If users move beyond the soft limit, they have a fixed time available (a grace period) to free up space by deleting files or blocks.

If users exceed the grace period, they cannot create any new files until they delete enough files to move under the soft limit.

Configure Grace Periods for Blocks and Inodes

You can edit the grace periods in vi for blocks and inodes by entering **edquota -t**. A screen similar to the following appears:

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem             Block grace period     Inode grace period
  /dev/sda2                     7days                  7days
```

You can set the grace periods in days, hours, minutes, or seconds for a listed file system. However, you cannot specify a grace period for a specific user or group.

Copy User Quotas

You can copy user quotas from one user to another by using **edquota -p**.

For example, by entering **edquota -p tux geeko**, you can copy the user quotas for the user **tux** to the user **geeko**.

Generate a Quota Report

The quota system files contain information in binary format about the space occupied by users and groups, and which quotas are set up. You can display this information by using the command repquota.

For example, entering **repquota -aug** displays a report similar to the following for all users and groups:

```
*** Report for user quotas on device /dev/sda2
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      -- 2646650       0       0         140161     0     0
geeko     +-   20000   10000   20000  7days     146     0     0
```

For additional details on using repquota, enter **man 8 repquota**.

Start and Activate the Quota Service

In order for the quota system to be initialized when the system is booted, the appropriate links must be made in the runlevel directories by entering **insserv quota** (**innserv quotad** for NFS).

You can then start the quota system by entering **/etc/init.d/quota start**.

You can also start or stop the quota system by entering the following:

usr/sbin/quotaon *filesystem*
usr/sbin/quotaoff *filesystem*

You can use the option -a to activate and deactivate all automatically mounted file systems (except NFS) with quotas.

For additional information on quotaon options, enter **man quotaon**.

Exercise 3-4 Set Up and Configure Disk Quotas

Do the following:

1. From a terminal window, su to root (**su -**) with a password of **novell**.
2. View the disk quota configuration for user geeko by entering the following:

   quota -vu geeko

   Notice that there are no quotas currently configured for geeko.
3. Add quota mount options to the partition /dev/hda6 by doing the following:
   a. Open the /etc/fstab file in the vi editor by entering **vim /etc/fstab**.
   b. Edit the /dev/hda6 entry to reflect the following:

      /dev/hda6 /export/data2 ext3 defaults**,usrquota, grpquota** 1 2
   c. When you finish, save the file and exit by entering **:wq**.
4. Remount the file system so it that reads the changes in file /etc/fstab by entering the following:

   mount -o remount /dev/hda6

If you receive an error message "/export/data2 not mounted already, or bad option" check the contents of the /etc/fstab file. You might have misspelled the usrquota or grpquota option.

5. Run quotacheck to initialize the quota database by entering the following:

   quotacheck -mavug

   You receive several status messages about old quota files. These indicate that this is a new quota database with no previous quota database files on the system.

6. Verify that the files aquota.user and aquota.groups exist in the directory /export/data2 by entering **ls -l /export/data2**.

7. Turn quotas on for all file systems that are mounted with these options by entering the following:

   quotaon -av

8. Make the quota system persistent after reboot by entering **chkconfig quota on**.

9. View the quota report by entering the following:

   repquota -av

   The quotas are set by using the number of 1k blocks.

   Notice that root is the only user listed.

10. Determine the block size on the file system by entering the following:

    dumpe2fs /dev/hda6 | more

    Enter the block size below:

11. Set a quota for geeko of a soft limit of 20 MB and a hard limit of 30 MB on /dev/hda6 by entering the following:

    edquota -u geeko

    The quota editor appears (the vi editor).

12. Enter the required ***soft limit*** and ***hard limit*** under the Soft and Hard columns for /dev/hda6 (press **Insert** twice to replace text).

    You can calculate the soft limit and hard limit sizes you need to enter by dividing the number of bytes by the block size.

    For example, if the block size you recorded is 4096, you would divide **20,000,000** by **4096** to calculate the soft limit number you need to enter.

You can access a calculator from the KDE Menu by selecting **Utilities > Calculator**.

13. When you finish, press **Esc**; then enter **:wq**.
14. Run repquota to view the quota information about all configured users by entering the following:

    repquota -av

    Notice that geeko is now listed with the soft limit and hard limit values you entered.
15. (Optional) If you finish early then set a quota for the users group of 100 MB for the soft limit, and 150 MB for the hard limit.
16. Close all open windows.

(End of Exercise)

Objective 6 Back Up and Restore the File System

Backing up and restoring system data is one of the most critical tasks performed by a system administrator. YaST provides backup and restore modules to help you with this task.

To back up and restore a file system on SUSE Linux Enterprise Server, you need to know the following:

- Data Backup Strategies
- Back Up System Data With YaST
- Restore System Data With YaST
- Data Backup Command Line Tools

Data Backup Strategies

The following are strategies and guidelines for implementing backup on your SUSE Linux system:

- **Backup media.** Because large amounts of data are normally included in a system backup, you first need to decide which media you want to use to back up the data.

  Administrators normally use tape drives, as these have the best price-to-capacity ratio and are SCSI drives. In addition, tapes have the advantage of being relatively simply to reuse.

  Other media for data backup include writable CDs or DVDs, removable hard drives and Magnetic-Optical (MO) drives.

  Storage Area Networks (SANs) are networks that back up data from different computers on a central backup server. But even SANs often use tape drive units to perform the backup.

- **Backup requirements.** When organizing data backups, you often need to compromise between the different requirements.

For example, lost data should be reconstructed as quickly as possible. However, the amount of data to be backed up should be kept as small as possible (only data that has changed since the last backup).

- **Backup frequency.** How often a backup is performed depends on the importance of the data.

  If the data is highly sensitive, then a complete daily backup is unavoidable. With less sensitive data, you can normally perform a weekly backup.

- **Tape availability.** You need several tapes that can be overwritten in a rolling backup process.

  You should keep 2 sets of tapes—1 for less sensitive data, and 1 for daily backup of sensitive data.

  For a daily backup, all the tapes for a week should be kept longer, which means at least 10 to 15 sets of tapes.

  If necessary, you can carry out incremental backups at regular intervals, that back up all data that has changed since the last complete backup.

- **Tape storage.** You should always store backup tapes separately from the server. This prevents backups from being lost in a disaster (such as fire in the server room).

  This is especially true of sensitive data, which is often kept in a separate, secure room, or even a secure bank vault.

Back Up System Data With YaST

The YaST System Backup module lets you create a backup of your system. The backup does not comprise the entire system, but only saves information about changed packages and copies of critical storage areas and configuration files.

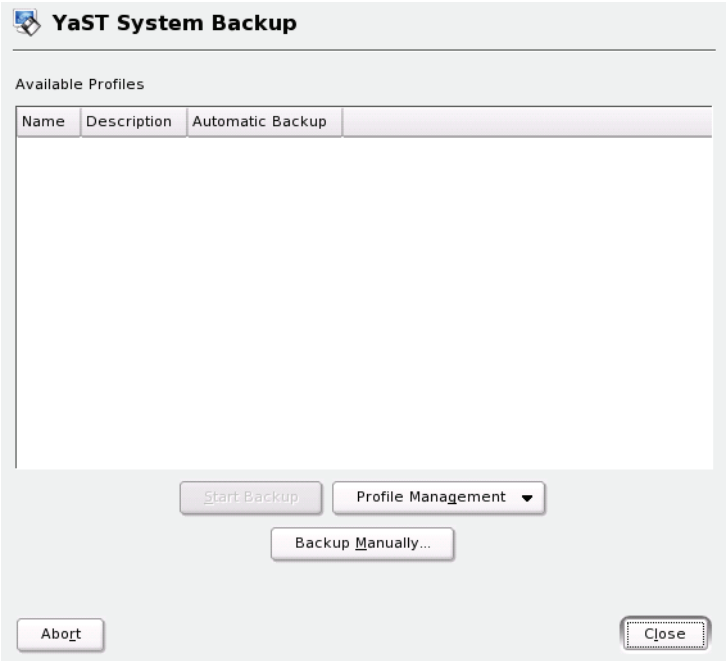
To create a backup with YaST, do the following:

1. From the KDE desktop, start the YaST System Backup module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > System Backup**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 backup**.

The following appears:

Figure 3-21



This dialog shows the list of currently stored backup profiles. A *backup profile* is used to name a group of different settings, such as name of an archive and how to search for files.

You can have a number of profiles, each with a unique name.

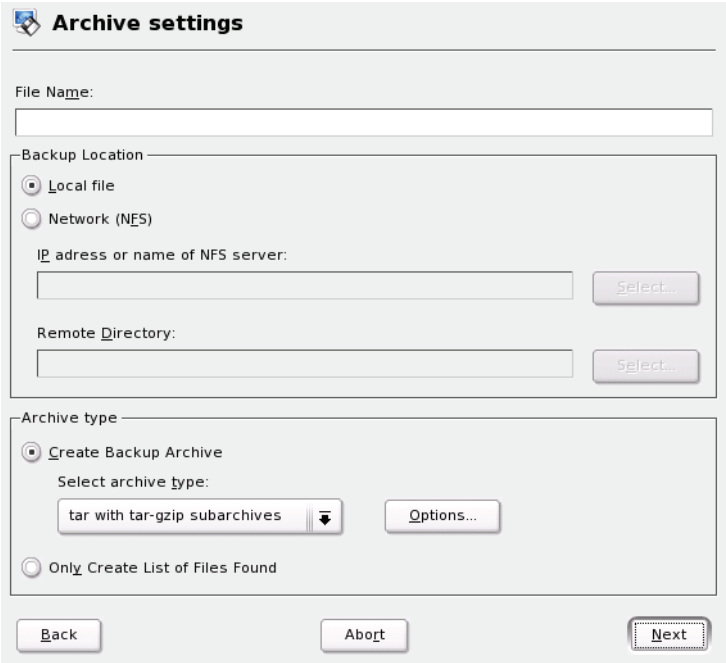
From the Profile Management drop-down list, you can add a new profile (**Add**) based on default values, duplicate an existing profile (**Duplicate**), edit the settings stored in a profile (**Change**), delete a profile (**Delete**), or configure automatic backup settings.

You can also use **Backup Manually** to configure a backup without creating a backup profile.

2. Create a profile by selecting **Profile Management > Add**.
3. Enter a ***name*** for the profile that will be used in the profile list; then select **OK**.

   The following appears:

Figure 3-22



4. In the **File Name** field, enter a ***filename*** for the backup file.

   You need to enter a full path (absolute path) with the filename (such as **/etc/backup_1**).
5. Save the backup file to a local directory by selecting **Local file**, or save the backup file to a remote server by selecting **Network (NFS)** and entering the remote server and directory.

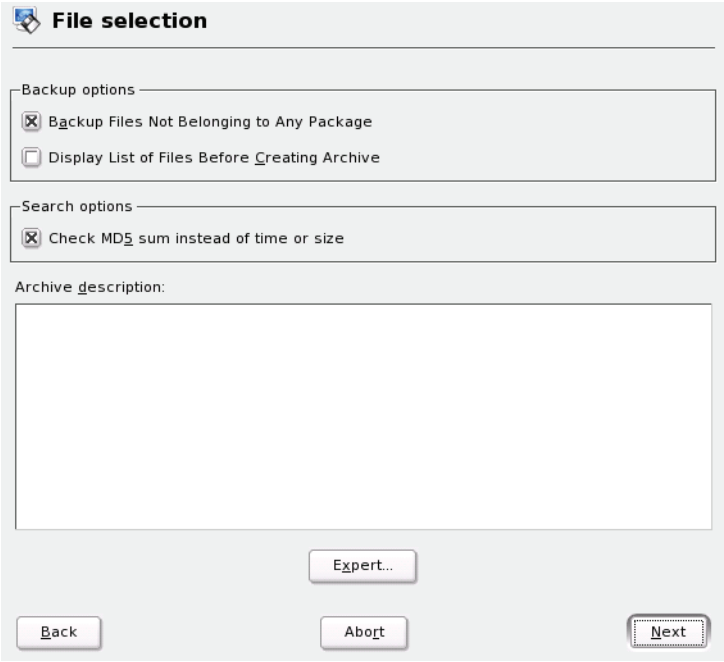
6. Create a backup file that contains the backup data by selecting **Create Backup Archive**, or select **Only Create List of Files Found**.

   The Create Backup Archive option lets you select an archive type (such as **tar with tar-gzip**) from a drop-down list, and configure additional options (such as multivolume archive) by selecting **Options**.

7. When you finish configuring the archive settings, continue by selecting **Next**.

   The following appears:

Figure 3-23



From this dialog you can select which parts of the system to search and back up.

The archive will contain files from packages that were changed since package installation or upgrade.

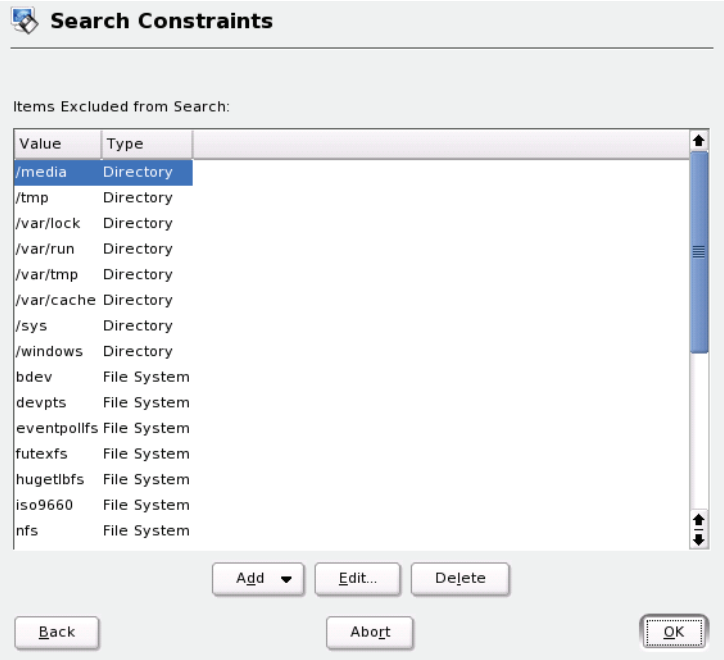
8. Select one or both of the following options:
   - **Backup Files Not Belonging to Any Package.** Include these files in the backup.
   - **Display List of Files Before Creating Archive.** Lets you show and edit a list of files found before creating the backup archive.
9. (Optional) In the **Archive Description** field, enter a ***description*** of the backup archive.
10. Use MD5 sum checking by selecting **Check MD5 sum instead of time or size**.

    You can use MD5 sum to determine if the file was changed. It is more reliable than checking size or modification time, but takes more time.
11. (Optional) Configure advanced options (such as adding the partition table to the backup) by selecting **Expert**.

    For most backups, you do not need to change the default Expert options.
12. When you finish configuring, continue by selecting **Next**.

The following appears:

Figure 3-24



This dialog lists all the items you want excluded from the backup, including the following exclusion types:

- **Directories.** All files located in the specified directories will not be backed up.
- **File Systems.** You can exclude all files located on a certain type of file system (such as ReiserFS or Ext2). The root directory will always be searched, even if its file system is selected.

  File systems that cannot be used on a local disk (such as network file systems) are excluded by default.

- **Regular expressions.** Any filename that matches any of the regular expressions will not be backed up. Use perl regular expressions. For example, to exclude *.bak files, add the regular expression **\.bak$**.

13. Add an item to the exclusion list by selecting **Add > exclusion type** and entering a ***directory***, ***file system***, or ***expression***; then select **OK**.
14. Edit or remove an item from the list by selecting the ***item***; then select **Edit** or **Delete**.
15. When you finish, continue by selecting **OK**.

    You are returned to the YaST System Backup dialog where the new profile appears in the list.
16. Start the backup by doing one of the following:
    - Select the profile; then select **Start Backup**.
    - Set an automatic backup by selecting **Profile Management > Automatic Backup**.

      You can set options such as backup frequency, backup start time, and maximum number of old backups.
17. When you finish configuring system backups, select **Close**.

Restore System Data With YaST

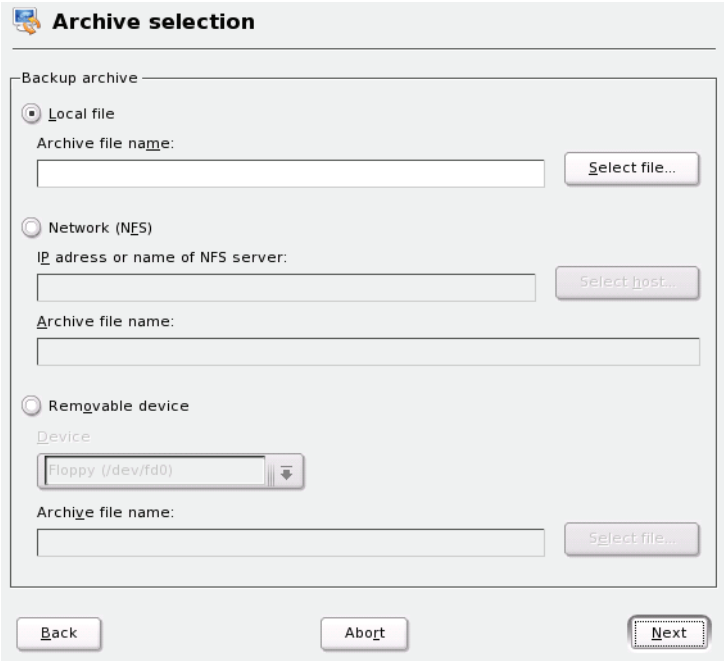
You can use the YaST Restore system module to restore a system backup by doing the following:

1. From the KDE desktop, start the YaST Restore system module by doing one of the following:
    - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > Restore system**.

      or
    - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 restore**.

The following appears:

Figure 3-25



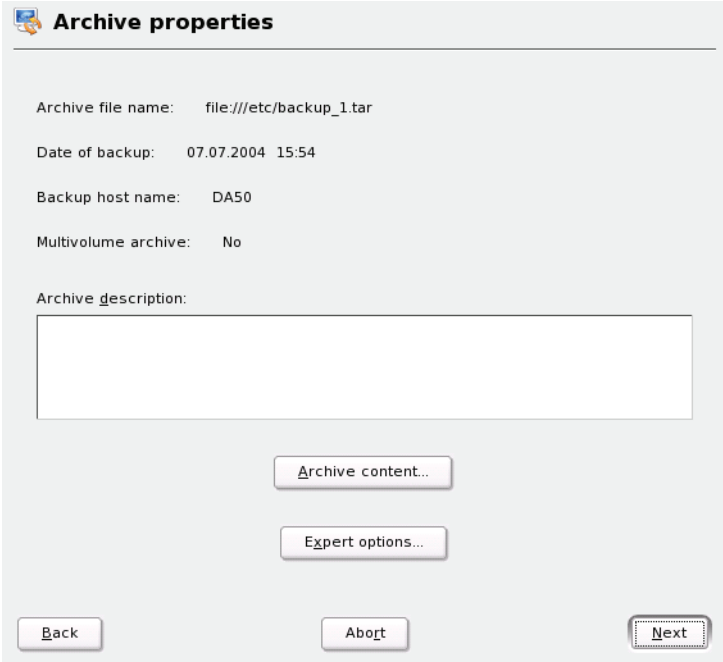
2. Do one of the following:
   - If the backup file is stored locally, select **Local file**; then enter the ***archive filename*** (include the full path) or locate and enter the file by selecting **Select file**.

     or
   - If the backup file is stored on a network server, select **Network (NFS)**; then enter the ***remote server*** and the full path of the ***archive backup file***.

     or
   - If the backup file is on a removable device (such as a diskette or tape drive), select **Removable device**; then select the ***device*** from the drop-down list and enter the full path of the ***archive backup file*** (or use **Select file**).
3. When you finish, continue by selecting **Next**.

YaST reads the contents of the archive file and the following appears:

Figure 3-26



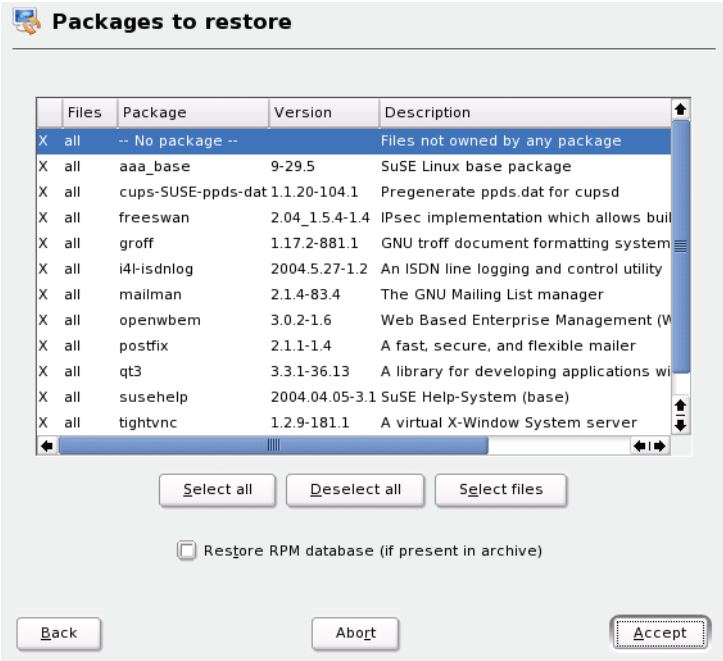
This dialog lists the properties of the archive file.

4. View the archive contents by selecting **Archive content**.
5. Configure options such as activating the boot loader configuration after restoration and entering the target directory by selecting **Expert Options**.
6. When you finish, continue by selecting **Next**.

If this is a multivolume archive, selecting Next displays an Archive properties dialog for each volume.

The following appears:

Figure 3-27



This dialog lets you select which files you want restored from the archive (all are selected by default).

The first column in the list displays the restoration status of the package. It can be **X** (package will be restored), empty (package will not be restored), or **P** (package will be restored partially).

The number of selected files that will be restored from the archive is in the second column.

Press Select Files to restore a package partially.

7. Do one of the following:
   - Select all packages in the list by selecting **Select all**.

     or
   - Deselect all packages in the list by selecting **Deselect all**.

     or
   - Restore particular files in a highlighted package by selecting **Select files**; then select or deselect the listed files.
8. (Optional) If the RPM database exists in the archive, restore it by selecting **Restore the RPM database**.
9. When you finish selecting packages, start restoring files by selecting **Accept**.

   When the restoration is complete, a summary dialog appears listing the status of the restored files.
10. (Optional) Save the summary to a file by selecting **Save to file**.
11. Close the dialog by selecting **Finish**.

Data Backup Command Line Tools

Linux has several command line tools available for data backup, including the following:

- **tar (tape archiver).** This is the most commonly used tool for data backup. It archives files in a special format directly on a corresponding medium (such as magnetic tape or formatted floppy disk) or to an archive file.

  Normally though, the data is not compressed. By convention, names of archive files end in .tar. If archive files are compressed (usually with the command gzip), then the extension of the filename is either .tar.gz or .tgz.

The command first expects an option (which is why it can also be used without a minus sign), then the name of the archive to be written (or the device file) and the name of the directory to be backed up.

All directories and files beneath this are also saved. Directories are typically backed up with a command similar to the following:

tar -cvf /dev/st0 /home

For additional details on the tar command, enter **man tar**.

- **rsync (remote synchronization).** The command rsync creates copies of complete directories across a network to a different computer.

  However, rsync can also be used to carry out local mirroring of directories. Only those files are copied that are not already in the target directory, or only exist in older versions. In fact, only parts of a file are copied that have changed, not the entire file.

  For example, the mirroring of all home directories can be carried out by entering a command similar to the following:

  rsync -a /home /shadow

  In this example, the mirroring is made to the directory **/shadow/**. There the directory **/home/** is first created, and then below it, the actual home directories of the users.

  If instead, you want the home directories created directly beneath the target directory specified (such as **/shadow/geeko/**), then you would enter the following command:

  rsync -a /home/. /shadow

  Specifying **/.** at the end of the directory to be mirrored indicates that this directory is not included in the copy.

The option **-a** switches rsync to the archive mode. This is a combination of other options (-rlptg), which ensures that the characteristics of the files to be copied are identical to the originals.

For additional details on rsync, enter **man rsync**, refer to the technical reference documentation in /usr/share/doc/packages/rsync/tech_report.ps, or connect to the rsync project web site at http://rsync.samba.org/.

- **dd command.** With the command dd, you can convert and copy files byte-wise.

  Normally dd reads from the standard input and writes the result to the standard output. However, with the correct parameters, however, you can also address files directory.

  You can copy all kinds of files with this command, including device files, which means entire partitions. Exact copies of an installed system (or just parts of it) can be created very simply.

  In the simplest case, you can copy a file with a command similar to the following:

  ```
  dd if=/etc/protocols of=protocols.org
  ```

  With the option **if=** (input file) you specify the file to be copied. With the option **of=** (output file) you specify the name of the copied file.

For additional details on using dd to copy, convert, or format files, see **man dd**.

- **mt command.** To work with magnetic tapes, Linux provides the command **mt**.

  With this command, you can position tapes, switch compression on or off (with some SCSI-2 tape drives), and query the status of the tape.

Magnetic tape drives in Linux are always SCSI devices and can be addressed by names such as the following:

- **/dev/st0:** refers to the first tape drive.
- **/dev/nst0:** addresses the same tape drive in the no rewind mode. In other words, after writing or reading, the tape remains at that position and is not rewound back to the beginning.

For compatibility with other Unix conversions, two symbolic links exist: **/dev/rmt0** and **/dev/nrmt0**.

You can view the status of a tape by entering a command similar to the following:

```
mt -f /dev/st0 status
```

For additional details on use the mt command, enter **man mt**.

- **cron service.** Backing up data is a task that should be carried out regularly. You can automate backup from the command line with the cron service.

  System jobs are controlled with the file /etc/crontab and the files in the directory /etc/cron.d/. Other jobs are defined with the scripts in the directories /etc/cron.hourly/, /etc/cron.daily/, /etc/cron.weekly/ and /etc/cron.monthly/.

  You specify which users can create cron jobs by using the files /var/spool/cron/allow and /var/spool/cron/deny, which are accessed in this order. If both files do not exist, then only root may define jobs.

  The jobs of individual users are stored in files in the directory /var/spool/cron/tabs/ with names matching the user names.

  You process these files by entering the command **crontab** (see **man crontab**).

The following is an example of a defined cron job:

```
0 22 * * 5 /root/bin/backup
```

The script /root/bin/backup is started every Friday at 10 o'clock in the evening. The format for the line is described in **man 5 crontab**.

Exercise 3-5 Back Up System Files on Your Server

In this exercise, you use the utilities gzip/gunzip, bzip2, and tar as archiving tools to back up files such as aafire (which displays burning flames) and user and group files.

Do the following:

1. From a terminal window, su to root (**su -**) with a password of **novell**.
2. Back up your aafire binary by copying it to root's home directory with the following command:

   cp /usr/bin/aafire ~
3. Check the size of the binary by entering **ls -lh aafire**.

   Record the size below:
4. Archive the binary file in a compressed format by entering **gzip -1 aafire**.
5. Check the size of the binary file by entering **ls -lh aafire.gz**.

   Notice that the file is smaller. Record the size:
6. Uncompress the file aafire.gz by entering **gzip -d aafire.gz**.
7. Compress aafire again, but this time use the highest compression by entering **gzip -9 aafire**.
8. Check the size of the binary by entering **ls -lh aafire.gz**.

   Record the size:

9. Uncompress the file aafire.gz by entering **gunzip aafire.gz**.

10. Compress the file aafire with bzip2 and check the file size by entering the following 2 commands:

    bzip2 aafire
    ls -lh aafire.bz2

11. Uncompress the file aafire.bz2 and check the file size by entering the following 2 commands:

    bzip2 -d aafire.bz2
    ls -lh aafire

12. Create a usrgrp.tar archive of /etc/passwd, /etc/shadow, /etc/group by entering the following:

    tar -cvf usrgrp.tar /etc/passwd /etc/shadow /etc/group

13. List the files in the tar archive by entering the following:

    tar tvf usrgrp.tar

14. Now create a gzip archive of the file usrgrp.tar and check the file size by entering the following 2 commands:

    gzip -9 usrgrp.tar
    ls -lh usrgrp.tar.gz

15. Extract the file usrgrp.tar.gz in a /tmp/ directory and check the file information by entering the following commands:

    mkdir tmp
    cd tmp/
    tar zxvf ../usrgrp.tar.gz
    ls -Rl

16. Verify you are in the directory /root/tmp by entering **pwd**.

17. (Conditional) If you are not in the directory /root/tmp, change to the directory by entering **cd /root/tmp**.

18. Delete all files in the directory tmp/ and check the results by entering the following:

    rm -rf *
    ls

19. Extract the archive again, but manually perform the same function as the command tar zxvf by entering the following:

    gzip -dc ../usrgrp.tar.gz | tar xvf -

20. Verify that the files were created by entering **ls -Rl**.

21. Close the terminal window.

(End of Exercise)

Summary

Objective	Summary
1. Select a Linux File System	One of the key roles performed by the Linux operating system is providing storage services through creating and managing a file system. To successfully select a file system that meets your server requirements, you learned the following about file systems available for Linux: ■ Linux File Systems ■ Linux File System Formats ■ Linux File System Characteristics ■ File System Journaling ■ Additional File System Documentation Each file system has its particular strengths and weaknesses, which must be taken into account. However, even the most sophisticated file system cannot be a substitute for a reasonable backup strategy.

Objective	Summary
2. Configure Linux File System Partitions	A basic task of all system administrators is maintaining file system layouts. Under Linux (and UNIX), new partitions can be transparently grafted into existing file system structures using the **mount** command. In most cases, YaST proposes a reasonable partitioning schema during installation that can be accepted without change. However, you can also use YaST to customize partitioning after installation. To implement partitions on your SUSE Linux Enterprise Server, you learned about the following: ■ Partition Types ■ Linux Device and Partition Names ■ Design Guidelines for Implementing Partitions ■ Design Guidelines for Optimizing Partitions ■ How to Manage Partitions With YaST

Objective	Summary
3. Configure a File System With Logical Volume Management (LVM)	Logical volume management (LVM) provides a higher-level view of the disk storage on a computer system than the traditional view of disks and partitions. When you create logical volumes with LVM, you can resize and move logical volumes while partitions are still mounted and running. To configure a file system with LVM, you learned about the following: ■ LVM Components ■ LVM Features ■ How to Configure Logical Volumes With YaST
4. Configure and Manage a Linux File System	To perform basic Linux file system management tasks in SUSE Linux Enterprise Server, you learned how to do the following: ■ Create a File System From YaST ■ Create a File System From the Command Line ■ Mount a File System ■ Monitor and Check a File System ■ Create a Boot, Rescue, or Module Disk

Objective	Summary
5. Set Up and Configure Disk Quotas	Linux includes a quota system that lets you specify a specific amount of storage space for each user or group, and how many files that user or group can create. In SUSE Linux Enterprise Server you can use the quota package to establish these limitations. In this objective, you learned how to perform the following quota management tasks: ■ Prepare the File System ■ Initialize the Quota System ■ Configure and Manage User and Group Quotas ■ Start and Activate the Quota Service
6. Back Up and Restore the File System	Backing up and restoring system data is one of the most critical tasks performed by a system administrator. YaST provides backup and restore modules to help you with this task. To back up and restore a file system on SUSE Linux Enterprise Server, you learned about the following: ■ Data Backup Strategies ■ Back Up System Data With YaST ■ Restore System Data With YaST ■ Data Backup Command Line Tools

SECTION 4 Manage Software for SUSE Linux Enterprise Server

In this section, you learn how to manage software packages on your SUSE Linux Enterprise server with RPM Package Manager (RPM) and YaST. You are also introduced to dynamic software libraries.

Objectives

1. Manage RPM Software Packages
2. Verify and Update Software Library Access
3. Manage Software Updates With YaST Online Update Server (YOU)

The Novell ZENworks Linux Management product (ZLM) is now available for managing software installation and updates on Linux servers and desktops in your network.

In addition to the upgrade method of using a remote YOU server, Novell provides upgrades to licensed and registered SLES 9 servers through a channel to a Novell ZLM server.

Although not included as part of the Novell CLP curriculum (or practicum testing), the ZLM client is available in SLES 9 for installation. For details on installing the client and subscribing to the upgrade channel, see *Appendix D: Novell ZENworks Linux Management (ZLM)*.

Objective 1 Manage RPM Software Packages

While there are several software package formats available for Linux, the format used most commonly in SUSE Linux installations is the RPM Package Manager (RPM) format.

Installing software in the RPM format can be done with YaST or by using the command rpm. YaST ensures the automatic resolution of dependencies, while rpm only controls them (resolution must be performed manually).

To manage installation of RPM software packages, you need to know the following:

- RPM Components and Features
- RPM Basics
- How to Manage Software Packages With rpm
- How to Manage Software Packages With YaST

RPM Components and Features

RPM Package Manager (or RPM) is a package management system primarily intended for Linux. RPM installs, updates, uninstalls, verifies and queries software.

The following are the basic components of RPM:

- **RPM Package Manager.** The utility that handles installing and uninstalling RPM packages.
- **RPM database**. The RPM database works in the background of the package manager and contains a list of all information for all installed RPM packages.

The database keeps track of all files that are changed and created when a user installs a program. This helps the package manager easily remove the same files that were originally installed.

If the database becomes corrupted, double links in the database make sure that it can normally be rebuilt without any trouble.

- **RPM package.** RPM lets you take software source code and package it into source and binary packages for end users. These are called RPM packages.
- **Package label.** Every RPM package includes a package label that contains information such as the software name, version, and the package release number.

  This information helps the package manager track the installed versions of software to make it easier to manage software installations on a Linux computer.

Some of the advantages of using RPM package manager and RPM packages include the following:

- Provides a consistent method for users to install programs in Linux.
- Makes it easier to uninstall programs (because of the RPM database).
- Most software packages for Linux (including those from Novell for Linux platforms) are now formatted as RPM packages.
- Original source archives (such as tar.gz, .tar.bz2) are included and easy to verify.
- You can use RPM tools to enable software installations using noninteractive scripts.
- You can use RPM tools to verify that the software installed correctly.
- RPM can track dependent software, which means that any additional software needed is also installed.

- RPM allows for all packaged software to use public-key technology to digitally sign the software.

RPM Basics

To manage software packages with RPM, you need to understand the following:

- RPM Package File Format
- RPM Configuration File
- RPM Database

RPM Package File Format

RPM package files use the following naming format:

software_name-software_version-release_number.architecture.rpm

The following describes each component of the naming format:

- ***software_name.*** This is normally the name of the software being installed.
- ***software_version.*** This is the version number of the software in the RPM package and is normally a number.
- ***release_number.*** This is the number of times the package has been rebuilt using the same version of the software.
- ***architecture.*** This indicates the architecture the package was built under (such as i586, i686, athlon, ppc) or the type of package content.

  For example, if the package has an i586 architecture, you can install it on 32-bit Intel-compatible machines that are Pentium class or higher.

If the package has a noarch extension, it does not include any binary code.

No matter what the RPM software package name, all RPM packages include a binary header.

RPM Configuration File

The global RPM configuration file of the command rpm is /usr/lib/rpm/rpmrc. However, when the rpm command is updated, all changes to this file are lost.

To prevent this from happening, write the changes to file /etc/rpmrc (for the system configuration) or to file ~/.rpmrc (for the user configuration).

RPM Database

The files of the RPM database are stored 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 larger than expected, it is useful to rebuild the database by entering **rpm --rebuilddb**. Before doing this, make a backup of the old database.

The cron script suse.de-backup-rpmdb stored in /etc/cron.daily/ checks daily to see if there are any changes. If so, a copy of the database is made (packed with gzip) and stored in /var/adm/backup/rpmdb/.

The number of copies is controlled by the variable MAX_RPMDB_BACKUPS (default is 5) in /etc/sysconfig/backup.

The size of a single backup is approximately 5 MB for 1 GB in /usr.

How to Manage Software Packages With rpm

You can use the command rpm to manage software packages. This includes querying the RPM database for detailed information about the installed software.

The command provides the following modes for managing software packages:

- Installing, uninstalling, or updating software packages
- Rebuilding the RPM database
- Querying RPM bases or individual RPM archives
- Checking the integrity of packages

You can use the command rpmbuild to build installable RPM packages from pristine sources.

These packages contain program files to install and certain meta information used during installation by RPM to configure the software package. This same information is stored in the RPM database after installation for documentation purposes.

RPM archives normally have the extension .rpm.

To manage software packages with RPM, you need to know how to do the following:

- Verify Package Authenticity
- Install, Update, and Uninstall Packages
- Update Software With Patches
- Query Archives and the RPM Database

For a number of packages, the components needed for software development (libraries, headers, include files, etc.) have been put into separate packages. These development packages are only needed if you want to compile software yourself (such as the most recent GNOME packages).

These packages can be identified by the name extension -devel, such as the packages alsa-devel, gimp-devel, and kdelibs-devel.

Verify Package Authenticity

All SUSE Linux RPM packages are signed with the following GnuPG key:

```
1024D/9C800ACA 2000-10-19 SuSE Package Signing Key
<build@suse.de>
Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8
9C80 0ACA
```

You can enter the command **rpm --checksig *package_name*** (such as **rpm --checksig apache-1.3.12.rpm**) to verify the signature of an RPM package. This lets you determine whether the package originated from SUSE or from another trustworthy facility.

Verifying the package signature is especially recommended for update packages from the Internet.

The SUSE public package signature key is stored in the directories /root/gnupg/ and /usr/lib/rpm/gnupg/. Storing the key in /usr/lib/rpm/.gnupg/ lets normal users verify the signature of RPM packages.

Install, Update, and Uninstall Packages

To manage RPM software packages, you need to know how to do the following:

- Install an RPM Package
- Update an RPM Package
- Uninstall an RPM Package

Install an RPM Package

For most RPM packages, you use the following command to install the software:

rpm -i *package_name*.rpm

During installation, the RPM database ensures that no conflicts arise (such as a file belonging to more than 1 package). The package is installed only if its dependencies are fulfilled and there are no conflicts with other packages.

If there is an error, RPM requests those packages that need to be installed to meet dependency requirements.

You use other options to ignore these defaults, but this is only for experts. If you don't know what you're doing, you can risk compromising the integrity of the system and possibly jeopardizing the ability to update the system.

Update an RPM Package

You can use the options -U (or --upgrade) and -F (or --freshen) to update a package by using the following syntax:

rpm -F *package_name*.rpm

This command removes the files of the old version and immediately installs the new files.

The difference between the 2 options is that -U installs packages that previously did not exist in the system, but -F simply updates previously installed packages.

RPM updates configuration files carefully using the following guidelines:

- If a configuration file was not changed by the system administrator, RPM installs 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 saves the changed file with the extension .rpmorig or .rpmsave (backup file). It then installs the version from the new package, but only if the originally installed file and the newer version are different.

  If this is the case, compare the backup file (.rpmorig or .rpmsave) with the newly installed file and make your changes again in the new file. Afterward, be sure to delete all .rpmorig and .rpmsave files to avoid problems with future updates.
- A set of .rpmnew files are created if the configuration file already exists and if the noreplace label was specified in the .spec file.

After an update, you need to remove .rpmsave and .rpmnew files (after comparing them) so they do not interfere with future updates.

The .rpmorig extension is assigned if the file has not previously been recognized by the RPM database; otherwise, .rpmsave is used.

In other words, .rpmorig results from updating from a foreign format to RPM. .rpmsave results from updating from an older RPM to a newer RPM.

.rpmnew does not disclose any information as to whether the system administrator has made any changes to the configuration file.

A list of these files is available in /var/adm/ rpmconfigcheck. Some configuration files (such as /etc/httpd/httpd. conf) are not overwritten to allow continued operation.

The option -U is *not* equivalent to uninstalling with the -e option and installing with the -i option. Use -U whenever possible for updating packages.

Uninstall an RPM Package

To uninstall (remove) an RPM package, enter the following:

rpm -e *package_name*

RPM will delete the package only if there are no unresolved dependencies. For example, it is theoretically impossible to delete Tcl/Tk if another application requires it.

Even in this case, RPM calls for assistance from the database. If such a deletion is impossible (even if no additional dependencies exist), it might be helpful to rebuild the RPM database using the option --rebuilddb.

Update Software With Patches

To guarantee the operational security of a system, you should update packages frequently by installing patches in the packages.

When planning an update, you need to consider the following (using the package **pine** as an example):

- Is the patch RPM suitable for my system?

  To check this, first query the installed version of the package:

  ```
  rpm -q pine
  pine-4.44-188
  ```

  The results indicate the currently installed version of pine. Then check if the patch RPM is suitable for this version of pine:

  ```
  rpm -qp --basedon pine-4.44-224.i586.patch.rpm
  pine = 4.44-188
  pine = 4.44-195
  pine = 4.44-207
  ```

  The results indicate that the patch is suitable for 3 different versions of pine. The installed version in the example is also listed, so the patch can be installed.

- Which files are replaced by the patch?

  The files affected by a patch can easily be seen in the patch RPM. The option -P lets you select special patch features.

  You can display the list of files with the following command:

  ```
  rpm -qpPl pine-4.44-224.i586.patch.rpm
  /etc/pine.conf
  /etc/pine.conf.fixed
  /usr/bin/pine
  ```

  If the patch is already installed, use the following command:

  ```
  rpm -qPl pine
  /etc/pine.conf
  /etc/pine.conf.fixed
  /usr/bin/pine
  ```

- How can a patch RPM be installed in the system?

  Patch RPMs are used just like normal RPMs. The only difference is that a suitable RPM must already be installed.

- Which patches are already installed in the system and for which package versions?

  You can display a list of all patches installed in the system with the command **rpm -qPa**. If only 1 patch is installed in a new system (as with pine) the following list appears:

  ```
  rpm -qPa
  pine-4.44-224
  ```

  If at a later date you want to know which package version was originally installed, you can query the RPM database.

  For pine, this information can be displayed with the following command:

  ```
  rpm -q --basedon pine
  pine = 4.44-188
  ```

For additional details about the patch feature of RPM, enter **man rpm** or **man rpmbuild**.

Query Archives and the RPM Database

With the -q option, you can inspect an RPM archive (by adding the option -p) and query the RPM database of installed packages.

The following are the most commonly-used RPM query options:

Table 4-1

Option	Results
-i	List package information
-l	Display a file list

(continued) **Table 4-1**

Option	Results
-f *file*	Find out to which package *file* belongs (the full path must be specified with *file*)
-s	Display a file list with status information (implies -l)
-d	List only documentation files (implies -l)
-c	List only configuration files (implies -l)
--dump	Display a file list with complete details (to be used with -l, -c, or -d)
--provides	List features of the package that another package can request with --requires
--requires, -R	List the capabilities the package requires
--scripts	List installation scripts (preinstall, postinstall, uninstall)

For example, entering the command **rpm -qi wget** displays the following information:

```
Name        : wget                          Relocations: (not relocatable)
Version     : 1.9.1                              Vendor: SuSE Linux AG,
Nuernberg, Germany
Release     : 45.3                         Build Date: Wed Jun 30 21:26:19
2004
Install date: Fri Jul 16 17:20:41 2004      Build Host: neumann.suse.de
Group       : Productivity/Networking/Web/Utilities   Source RPM:
wget-1.9.1-45.3.src.rpm
Size        : 1503802                           License: GPL
Signature   : DSA/SHA1, Wed Jun 30 21:32:17 2004, Key ID a84edae89c800aca
Packager    : http://www.suse.de/feedback
URL         : http://wget.sunsite.dk/
Summary     : A tool for mirroring FTP and HTTP servers
Description :
Wget enables you to retrieve WWW documents or FTP files from a server.
This can be done in script files or via the command line.


Authors:
--------
    Hrvoje Niksic <hniksic@srce.hr>
Distribution: SuSE SLES-9 (i586)
```

The option -f only works if you specify the complete filename with a full path. You can enter several filenames, as in the following:

```
rpm -q -f /bin/rpm /usr/bin/wget
rpm-3.0.3-3
wget-1.5.3-55
```

This returns information for both /bin/rpm and /usr/bin/wget.

If you know only part of the filename, you can use a shell script to search for packages, as in the following:

```
#! /bin/sh
for i in $(rpm -q -a -l | grep $1); do
   echo "\"$i\" is in package:"
   rpm -q -f $i
   echo ""
done
```

Enter the partial filename when running the script.

Entering the command **rpm -q --changelog *rpm_name*** displays a detailed list of information (updates, configuration, modifications, etc.) about a specific package.

The following example displays information about the package rpm:

```
rpm -qp --changelog /media/cdrom/suse/i586/rpm-3*.rpm
```

Only the last 5 change entries in the RPM database are listed. All entries (dating back the last 2 years) are included in the package itself.

The rpm package query only works if CD 1 is mounted at /media/cdrom/.

With the help of the installed RPM database, you can perform verification checks with the option -V, -y, or --verify. All files in a package that have been changed since installation are displayed.

RPM uses the following character symbols to provide hints about the changes:

Table 4-2

Character	Description
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)

In the case of configuration files, the letter “c” is displayed. The following is an example for changes to /etc/wgetrc (wget):

```
rpm -V wget
S.5....T c /etc/wgetrc
```

Exercise 4-1 Manage Software With RPM

The package aalib is an ASCII art library that includes the file aafire which displays burning ASCII art flames.

In this exercise, you learn how to manage RPMs by working with the library aalib.

Do the following:

1. Use RPM to find out information about the package aalib:
   a. From a terminal window, determine which package installed the file /usr/bin/aafire by entering the following:

      rpm -qf /usr/bin/aafire

      Notice that the package aalib installed the file aafire.

   b. Find out information about the package aalib by entering the following:

      rpm -qi aalib

      Notice that the information includes the install date and a description.

   c. Show all the files installed by the package aalib by entering the following:

      rpm -ql aalib

      Where can you find information about the package aalib? (Notice the location of the README files.)

2. See what has changed in the files on your hard drive since the RPMs were originally installed by entering the following:

   rpm -Va

   Interpret the change information for each file listed by using Table 4-2.

3. When you finish viewing the information for a few files, stop the process by pressing **Ctrl+C**.

4. View the documentation files for the command at by entering the following:

   rpm -qd at

   Notice that some of the files are still compressed (*.gz).

5. (Optional) Try installing a package (**YaST > Software > Install and Remove Software**) from the SLES 9 Installation Server on DA1 and run **rpm -K *package*** on the package to verity the digital signature.

6. Close the terminal window.

(End of Exercise)

How to Manage Software Packages With YaST

To manage software packages with YaST, you need to know the following:

- How to Manage Software Packages from the GUI Interface
- How to Install a Package From the Command Line

How to Manage Software Packages from the GUI Interface

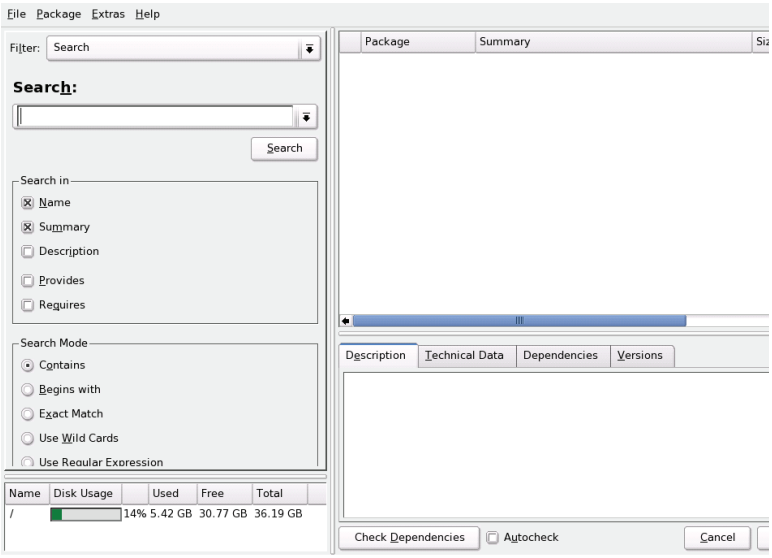
You can install, update, and remove (uninstall) software packages on your SUSE Linux Enterprise server with YaST by doing the following:

1. From the KDE desktop, start the YaST Install and Remove Software module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Software > Install and Remove Software**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 sw_single**.

The following appears:

Figure 4-1



This is the same YaST package manager dialog you use during installation to select software packages for a custom installation.

You can modify the frame sizes in the dialog by dragging the lines separating the areas.

2. From the Filter drop-down list, select one of the following to locate the package you want to install, update, or uninstall:
   - **Selections.** This filter groups the program packages according to their application purpose, such as multimedia or office applications.

- **Package Groups.** This filter provides a more technical overview of the range of packages and is suitable for users familiar with the package structure of SUSE Linux.

  The filter sorts the program packages by subjects, such as amusements, development, and hardware, in a tree structure to the left.

  To display all packages in alphabetical order, select **zzz All** in the top level.

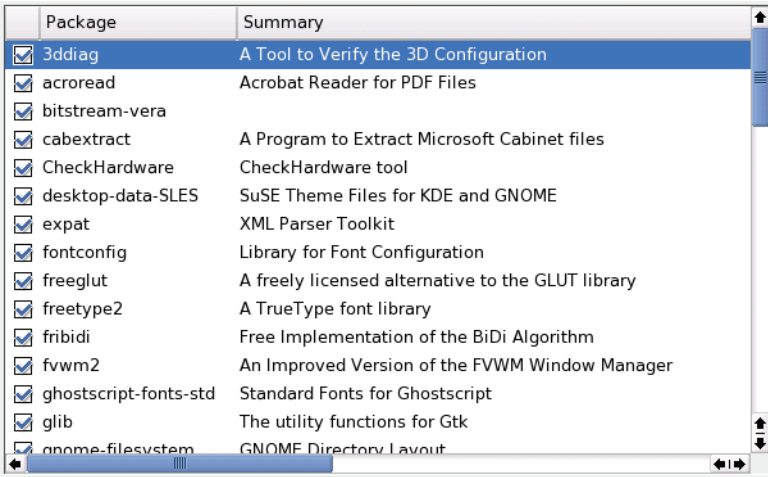
- **Search.** The Search feature is the easiest way to find a specific package.

  Enter a search string and use the check boxes to configure where to search for this string (in the name, in the description, or in the package dependencies).

  You can even define special search patterns using wildcards and regular expressions and search the package dependencies in the Provides and Requires fields.

Once you select a package, it is displayed in the individual packages window, as in the following:

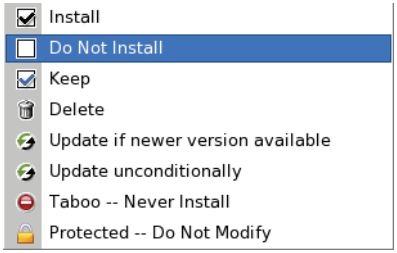
Figure 4-2



3. Right-click a ***package*** you want to install, update, or uninstall; then select **All in This List**.

   The following menu appears:

Figure 4-3



4. Select one of the following options:
   - **Install.** This package is not yet installed but will be installed.
   - **Do Not Install.** This package is not installed and will not be installed.
   - **Keep.** This package is already installed and will not be changed.
   - **Delete.** This package is already installed and will be deleted.
   - **Update if newer version available.** This package is already installed and will be replaced by the newer version on the installation medium.
   - **Update unconditionally.** This package is already installed and will be replaced by the version on the installation medium, whether or not it is a new version.
   - **Taboo — Never Install.** This package is not installed and will never be installed. It will be treated as if it does not exist on any of the installation media

If a package would automatically be selected to resolve dependencies, you can prevent this by setting Taboo. However, this can cause inconsistencies that must be resolved manually (dependency check).

- **Protected -- Do Not Modify.** This package is installed and should not be modified.

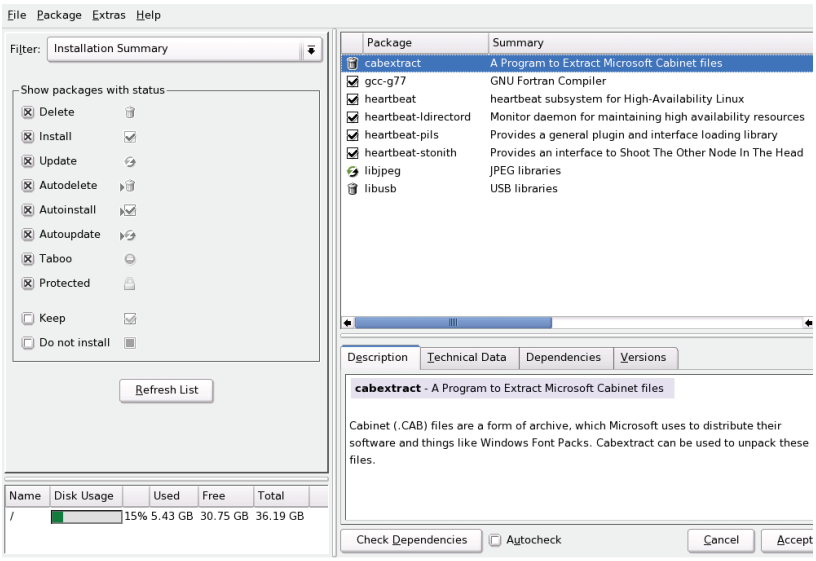
  Third-party packages (packages without the SUSE signature) are automatically assigned this status to prevent them from being overwritten by later versions that exist on the installation media.

  This can cause package conflicts that must be resolved manually.

5. (Optional) Locate and change the status of other packages you want to install or uninstall.
6. When you finish, from the Filter drop-down list select **Installation Summary**.

The following appears:

Figure 4-4



From this dialog you can see a list of all the packages you have modified and what will happen with the packages when you select Accept.

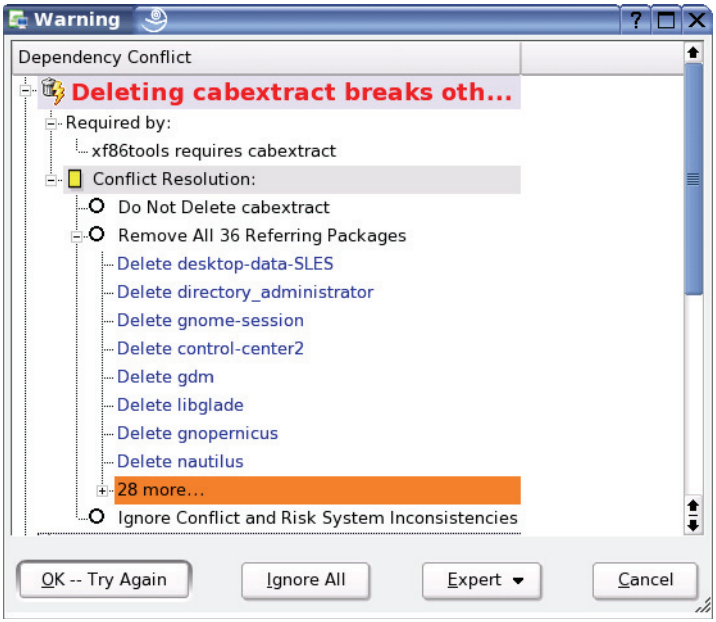
7. (Optional) Under **Show packages with status** (on the left), display only those packages with a particular status by selecting (or deselecting) the ***status***.

   Remember that only those packages displayed in the list will be processed by the package manager.

8. Display the dependencies for a particular package by selecting the package; then select Check Dependencies.

A dialog similar to the following appears:

Figure 4-5



The package manger checks for any unresolved package dependencies or conflicts. In the event of unresolved dependencies, the required additional packages are selected automatically.

For package conflicts, the package manager opens a dialog that shows the conflict and offers various options for solving the problem.

9. (Optional) If you want any change of a package status to trigger an automatic check, select **Autocheck**.

   Autocheck is a useful feature, as the consistency of the package selection is monitored permanently. However, this process consumes resources and can slow down the package manager.

For this reason, the autocheck is not activated by default. However, a consistency check is always performed when you confirm your selection with Accept.

10. When you finish viewing the packages, start the package manager by selecting **Accept**.

    The packages are installed, updated, or uninstalled.

    You are notified when any installation medium (such as a CD) needs to be accessed.

How to Install a Package From the Command Line

One of the major functions of YaST is software installation. If you know the name of a software package, the option -i (install) is very useful.

The following is an example:

```
yast -i ethereal
```

This example installs the ethereal package plus any software package that is needed by ethereal.

Exercise 4-2 Install a Software Package With YaST

SLES 9 includes a group of graphics utilities that are not included in a full installation.

In this exercise, you install these utilities from server DA1 by doing the following:

1. Verify that the graphics utilities are not installed by selecting the **KDE menu**.

   Notice that there is no **Graphics** menu item listed.

2. From the KDE Desktop, select the **YaST** icon; then enter a password of **novell** and select **OK**.

   The YaST Control Center appears.

3. Verify that the installation source is DA1:

   a. Select **Software > Change Source of Installation**.

      A Software Source Media dialog appears with a list of sources.

   b. Make sure that **SUSE SLES Version 9** and **SUSE CORE Version 9** sources point to 10.0.0.254 (DA1).

   c. Select the **Add** drop-down list and notice that additional installation sources are available to configure such as FTP, HTTP, and Samba.

      For example, if your installation server is not available, you can create a source that is configured for your CD-ROM drive mountpoint to install from CD.

   d. Close the dialog by selecting **Abort**; then select **Yes**.

4. From the YaST Control Center, select **Software > Install and Remove Software**.

   A package selector dialog appears.

5. In the Filter drop-down menu, make sure **Search** is selected.

6. In the Search field enter **graphics**; then select **Search**.

7. In the Packages list (to the right) select the following:
   - **kdegraphics3**
   - **kdegraphics3-extra**
8. Check dependencies to make sure the prerequisite packages are loaded by selecting **Check Dependencies**.

   A dialog appears indicating that all package dependencies are OK.
9. Close the dialog by selecting **OK**.
10. Install the packages by selecting **Accept**.

    The packages are installed and configured on your system from server DA1.
11. Verify that the graphics utilities have been installed by selecting the **KDE menu.**

    You see a new **Graphics** item on the menu.
12. From the KDE menu, select **Utilities > Desktop**.

    Graphic utilities such as KColorChooser, KColorEdit, and KSnapshot have been added to the menu.
13. Close the YaST Control Center.
14. (Optional) Find out more about the packages kdegraphics3 and kdegraphics3-extra by using the commands **rpm -qi** and **rpm -ql**.

(End of Exercise)

Objective 2 Verify and Update Software Library Access

In addition to checking for software package dependencies, you might also need to verify that the system is configured properly to access dynamic libraries an application uses.

Normally this is handled by the software installation, but occasionally you might need to verify software library access after installation.

For example, if an application that has been installed fails to start, try starting it from a terminal window. If the application reports that a library could not be found, then you might need to verify access to the dynamic libraries.

To verify the libraries needed for an application, you need to know the following:

- Software Library Basics
- How to View Shared Library Dependencies (ldd)
- How to Modify the Software Library Configuration File (/etc/ld.so.conf)
- How to Update the Library Cache (/etc/ld.so.cache)

Software Library Basics

To understand the role of software libraries in SUSE Linux, you need to know the following:

- Dynamic Software Libraries
- Static Software Libraries
- Library Naming Syntax

Dynamic Software Libraries

In a Linux environment, most programs share some code through the use of shared libraries. This provides advantages from a development and a system management standpoint.

For developers, it means their programs include only the code that is unique to the program itself, sharing functions that other programs have in common with it.

This reduces the size of the program executable, reducing the amount of disk space required for the application (an advantage for system administrators).

Unlike some other operating systems, a Linux system locates its dynamic libraries through a configuration file that points to the locations, eliminating confusion about which version of which dynamic library is used by each piece of software.

Developers still have the ability to link everything into their executable. This can be important if the program will be used on a system that might not include all of the necessary libraries, such as an emergency rescue disk or minimal Linux installation.

Static Software Libraries

In contrast to dynamic program linking, you can link the needed libraries statically when a program is compiled.

Although static linking increases the program size, it provides independence from libraries at runtime, and is especially useful for system maintenance purposes.

Programs with statically linked libraries include sash and insmod.

Library Naming Syntax

Library filenames normally use the following syntax:

`libname.so.`*`version`*

The letters “so” indicate a shared dynamic library; the letter “a” is used for static libraries. The version indicates a major version number of the library (such as 1, 2, or 6).

For example, the library used for the ncurses screen library (version 5.3) might be named:

`libncurses.so.5.3`

How to View Shared Library Dependencies (ldd)

You can view the shared libraries required by a specific program or shared library by using the command ldd.

The following is the syntax of the command:

ldd ***option filename***

For example, if you enter **ldd -v /opt/kde3/bin/suseplugger**, information similar to the following appears:

```
geeko@DA50:~> ldd -v /opt/kde3/bin/suseplugger
        linux-gate.so.1 =>  (0xffffe000)
        libhd.so.8 => /usr/lib/libhd.so.8 (0x40031000)
        libkio.so.4 => /opt/kde3/lib/libkio.so.4 (0x40158000)
        libkdeui.so.4 => /opt/kde3/lib/libkdeui.so.4 (0x40464000)
        libkdesu.so.4 => /opt/kde3/lib/libkdesu.so.4 (0x40706000)
        libkdecore.so.4 => /opt/kde3/lib/libkdecore.so.4 (0x4071f000)
        libDCOP.so.4 => /opt/kde3/lib/libDCOP.so.4 (0x4090b000)
        libresolv.so.2 => /lib/libresolv.so.2 (0x4093f000)
        libart_lgpl_2.so.2 => /usr/lib/libart_lgpl_2.so.2 (0x40951000)
        libkdefx.so.4 => /opt/kde3/lib/libkdefx.so.4 (0x40967000)
        libqt-mt.so.3 => /usr/lib/qt3/lib/libqt-mt.so.3 (0x40993000)
        libXi.so.6 => /usr/X11R6/lib/libXi.so.6 (0x4107d000)
        libXrandr.so.2 => /usr/X11R6/lib/libXrandr.so.2 (0x41085000)
        libXcursor.so.1 => /usr/X11R6/lib/libXcursor.so.1 (0x41089000)
        ...
```

For additional information on the command ldd, from a terminal window, enter **man ldd**.

How to Modify the Software Library Configuration File (/etc/ld.so.conf)

The file /etc/ld.so.conf contains a list of paths the Linux system uses to search for libraries, as in the following:

```
/usr/X11R6/lib/Xaw95
/usr/X11R6/lib/Xaw3d
/usr/X11R6/lib
/usr/i486-linux/lib
/usr/i486-linux-libc5/lib=libc5
/usr/i486-linux-libc6/lib=libc6
/usr/i486-linuxaout/lib
/usr/i386-suse-linux/lib
/usr/local/lib
/usr/openwin/lib
/opt/kde/lib
/opt/kde2/lib
/opt/kde3/lib
/opt/gnome/lib
/opt/gnome2/lib
include /etc/ld.so.conf.d/*.conf
```

In order to modify the file /etc/ld.so.conf, you need to be authenticated as the root user. The file format for this file is simply a list of system directories containing dynamic libraries.

Typical library directories include the following: /lib/, /usr/lib/, /usr/local/lib/, and /usr/X11R6/.

As the directories /lib and /usr/lib are taken into account in all cases, they are not listed in this file. You can enter the command **/sbin/ldconfig -p** to list all libraries available in the cache that will be found by the system.

If a library is located in a directory not listed above, you can set the variable LD_LIBRARY_PATH=*path* (as in the following) to make sure that it is loaded:

export LD_LIBRARY_PATH=*path*

For a listing of variables that can be used, enter **man 8 ld.so**.

How to Update the Library Cache (/etc/ld.so.cache)

The program ld.so or /lib/ld-linux.so.2 (this is a link to /lib/ld-2.2.3.so), referred to as the *runtime linker*, makes sure that the needed libraries are found and loaded when a program is started.

If you modify the /etc/ld.so.conf to reflect the new dynamic library paths, you need to enter the command ldconfig to update the library cache. If new libraries are installed during operation, you also need to enter ldconfig manually.

This is the same command used to update the library cache when rebooting the system.

The command sets the required links to the current shared libraries that are either located in the file /etc/ld.so.conf or in the directories /usr/lib/ and /lib/.

The library cache file is /etc/ld.so.cache and is read by the runtime linker. The cache file contains a list of all the system libraries stored in a binary format to speed the location of the libraries on the system.

If you need more information about the libraries found in each directory (such as a library is not being found when it is in one of the paths) you can run the command with the option -v to display detailed information about the libraries ldconfig has found.

The following is an example of using the option -v:

```
/lib/tls: (hwcap: 0x8000000000000000)
        libc.so.6 -> libc.so.6
        libpthread.so.0 -> libpthread.so.0
        libthread_db.so.1 -> libthread_db.so.1
        librt.so.1 -> librtkaio.so.1
        libm.so.6 -> libm.so.6
/lib/i686: (hwcap: 0x8000000000000)
        libc.so.6 -> libc.so.6
        libpthread.so.0 -> libpthread.so.0
        libm.so.6 -> libm.so.6
/usr/lib/tls: (hwcap: 0x8000000000000000)
        libdb_cxx-4.2.so -> libdb_cxx.so
        libdb-4.2.so -> libdb.so
```

Exercise 4-3 Manage Shared Libraries

In this exercise, you use some common utilities to manage the shared libraries on your SLES 9 server.

Do the following:

1. From a terminal or terminal window, su to root (**su -**) with a password of **novell**.
2. View the shared libraries:
   a. View all of the libraries linked to the SLP daemon (slpd) by entering **ldd /usr/sbin/slpd**.

      There are several libraries listed, including the file /lib/libnsl.so.1.

   b. Rename the /lib/libnsl.so.1 file to /lib/libnsl.so.1.bak by entering the following:

      mv /lib/libnsl.so.1 /lib/libnsl.so.1.bak

   c. Enter **ldd /usr/sbin/slpd** again.

      Notice that the link libnsl.so.1 indicates that the library is not found.

      By using the command ldd, you can find out if all required libraries are installed on a system for a specific program.

   d. Rename the file /lib/libnsl.so.1.bak back to /lib/libnsl.so.1 by entering the following:

      mv /lib/libnsl.so.1.bak /lib/libnsl.so.1

   e. Verify that the file can be found again by entering **ldd /usr/sbin/slpd**.
3. Rebuild the library cache:
   a. View the library cache by entering **ldconfig -p**.
   b. Rebuild the system library cache by entering **ldconfig -v**.

   The file /etc/ld.so.conf tells the Linux system where to look for libraries.

4. View the contents of the file /etc/ld.so.conf by entering **less /etc/ld.so.conf**.
5. When you finish viewing the contents, exit the display by typing **q**.
6. Close the terminal window.

(End of Exercise)

Objective 3 Manage Software Updates With YaST Online Update Server (YOU)

You can use the YaST module YOU Server Configuration to create a local update server for your network. The YOU server makes current software updates available to all YOU clients in the network.

To manage software updates with a YOU server, you need to know the following:

- YOU Basics
- How to Configure a Local YOU Server
- How to Configure and Use a YOU Client

YOU Basics

This YaST feature centralizes the updating of all systems in the network. The YOU server is compared either manually or automatically with one of the update servers in the Internet authorized by SUSE.

Depending on the product, these are either the SUSE maintenance web site (http://sdb.suse.de/download) or one of the mirrors of the SUSE FTP server (ftp://ftp.suse.com/pub/suse).

The local YOU clients start the updates via HTTP. The YOU server can be configured so that it can be recognized via SLP (Service Location Protocol) and can be recognized automatically by all clients in the network.

How to Configure a Local YOU Server

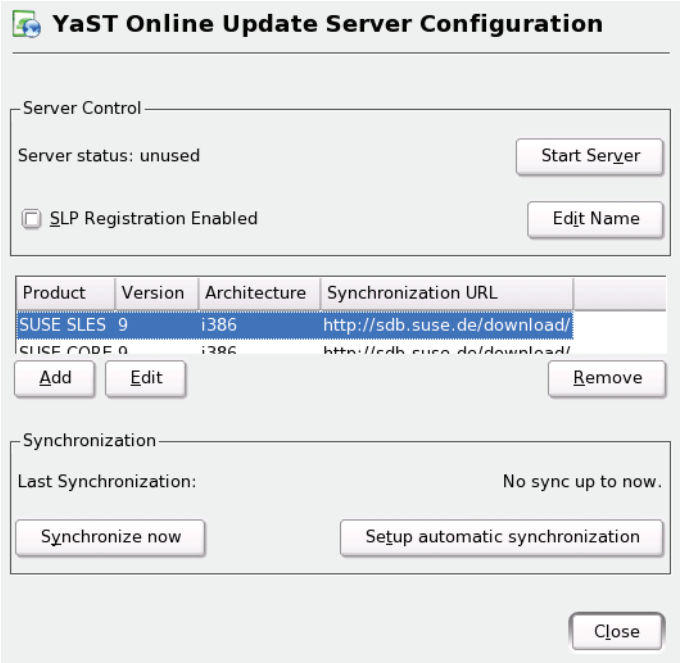
To configure a YOU server for your local network, do the following:

1. From the KDE desktop, start the YaST YOU Server Configuration module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Software > YOU Server Configuration**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 you_server**.

   The following appears:

Figure 4-6



From this dialog you can configure and control the YOU server.

2. Start the YOU server by selecting **Start Server**.

   The web server (apache2) that distributes the updates to the YOU clients via HTTP is installed, configured, and started.

   The **Server status** now indicates that the server is **running**.

3. (Optional) Register the YOU server as a service on the local SLP server:

   a. Name the SLP service by selecting **Edit Name**; then enter a name (such as **YOU Server**) and select **OK**.

   b. Select **SLP Registration Enabled**.

   The product list in the middle of the dialog shows the names of all products for which the YOU server currently provides updates and indicates the URL from which the updates are derived.

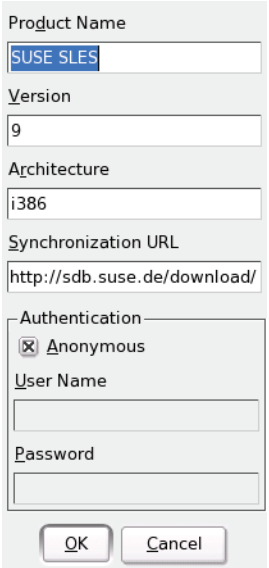
   The product running on the machine on which the server is set up is displayed as the default.

4. (Optional) Add or change a product in the list by doing the following:

   a. Select **Add** or **Edit**.

The following appears:

Figure 4-7



b. Enter or edit parameters such as the product name, hardware architecture, version, and URL of origin.

The product name, version, and architectural designation of the hardware are used internally by YOU to form the path under which the updates are searched for on the source server.

Make sure that you enter the correct information; otherwise, YOU cannot find the required updates on the source server.

---

c. (Conditional) In the case of update servers that require authentication (such as SUSE Maintenance Web), deselect **Anonymous**; then enter the user name and password.

d. When you finish, select **OK**.

5. Get the current patches for all products by selecting **Synchronize Now**.

   When the download is complete (and the download can take a long time), the date of the synchronization process is displayed.

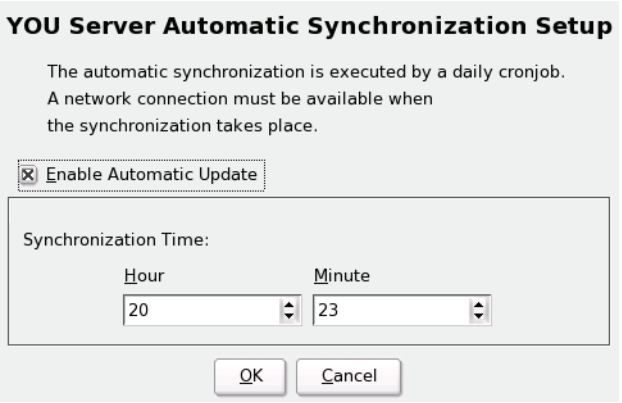
   All updates are stored in the directory /var/lib/YaST2/you/mnt/. From here the updates are available to all associated YOU clients for installation purposes.

6. (Optional) Set up automatic synchronization by configuring a daily cron job:

   a. Select **Setup automatic synchronization**; then select **Enable Automatic Update**.

      The following appears:

Figure 4-8



   b. Enter an ***hour*** and ***minute*** when you want the synchronization to start each day.

   c. Select **OK**.

7. When you finish setting up the YOU server, select **Close**.

How to Configure and Use a YOU Client

Installing software is not an absolutely static proposition. Developers find and fix bugs, security issues are resolved, and patches are created to remedy both.

To keep your SLES 9 system up-to-date and secure, you need to watch for and install relevant patches.

The solution for handling patches in SLES 9 is the is the YOU client. Although there is a YaST YOU module, you can also start the tool from the command line, from a script, or as a cron job.

To use YOU, you need a valid maintenance contract. As part of that contract you are issued a username and password to access the updates on SUSE servers.

To update your SLES 9 server with the YaST Online Update tool, you need to know the following:

- How to Install Patches From a Remote YOU Server
- How to Install Patches From a Local YOU Server
- How to Install Patches From the Command Line

How to Install Patches From a Remote YOU Server

To use YaST to update your SUSE Linux Enterprise Server system from a remote YOU server (such as the Novell SUSE Linux update server), do the following:

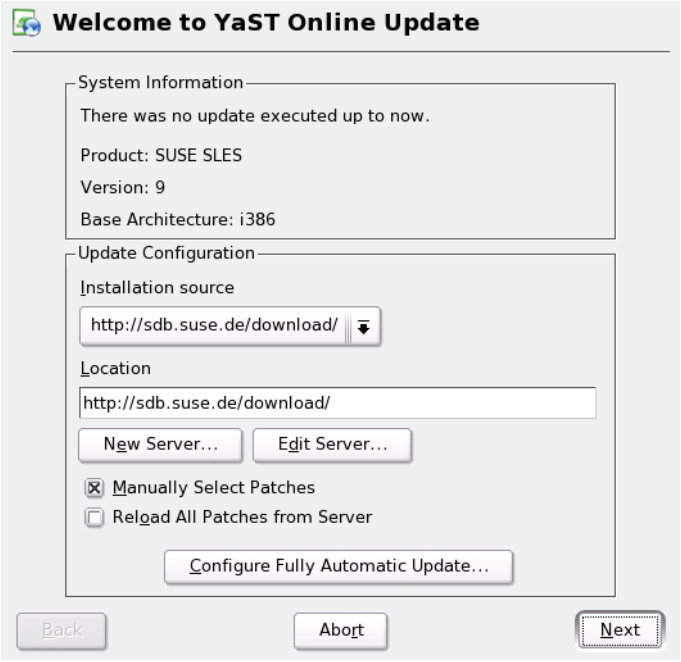
1. From the KDE desktop, start the YaST Online Update module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Software > Online Update**.

     or

- Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 online_update**.

The following appears:

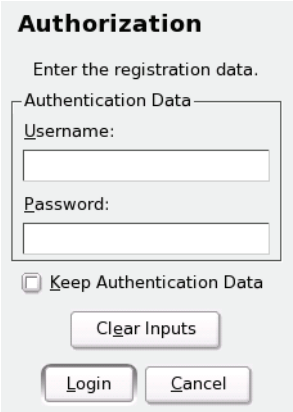
Figure 4-9



2. Accept the defaults and continue by selecting **Next**.

The following appears:

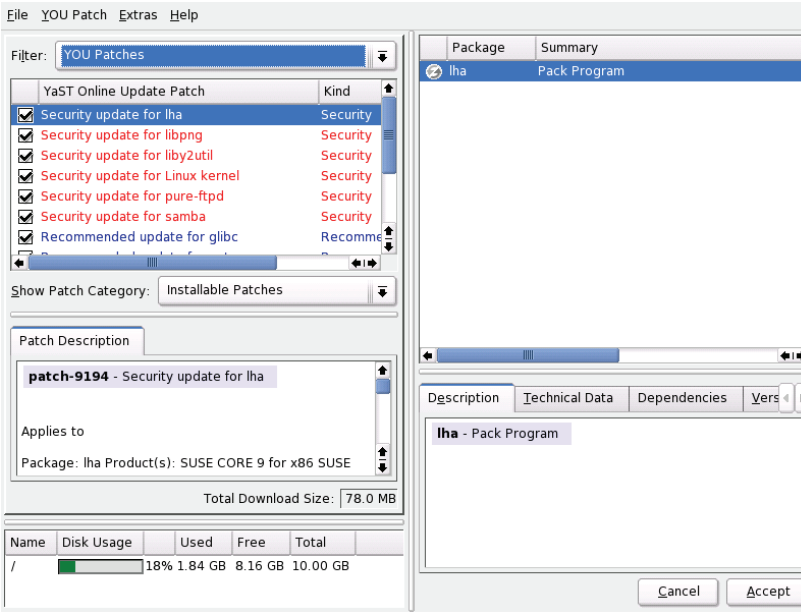
Figure 4-10



3. From your maintenance contract, enter your ***username*** and ***password***; then select **Login**.

The following appears.

Figure 4-11



From this dialog you can filter the patch list view and select or deselect the patches you want to install.

4. When you finish manually selecting or deselecting patches to install, select **Accept**.
5. When the download and installation is complete, select **Remove Source Packages after Update**.
6. Update the system configuration by selecting **Finish**.

How to Install Patches From a Local YOU Server

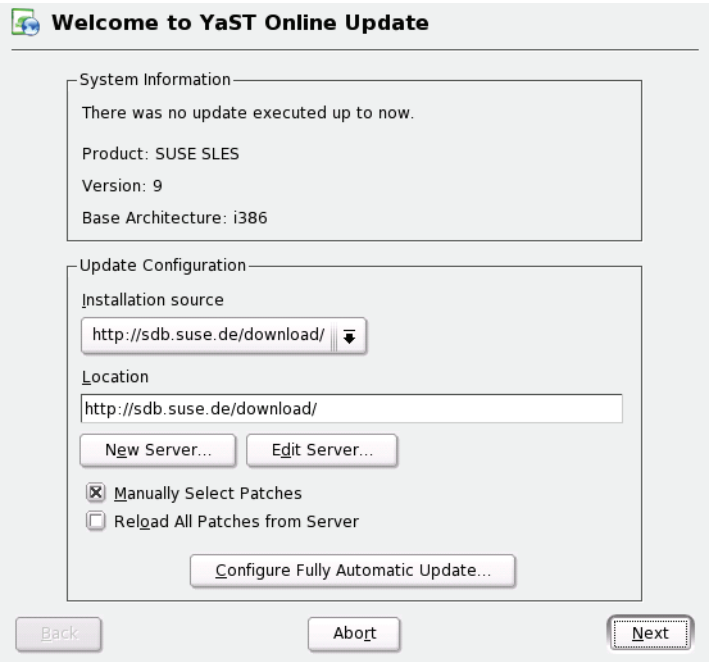
To configure a YOU client to access the YOU server on your local network, do the following:

1. (Conditional) If you registered the YOU server as an SLP service on the local SLP server, do the following:
   a. Open the file **/etc/sysconfig/onlineupdate** in a text editor (such as Kate).
   b. Scroll down to the SLP_ENABLED variable, and activate the SLP search on your YOU client by changing the value to **yes**.
   c. Save the change.
2. From the KDE desktop, start the YaST Online Update module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **Software > Online Update**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 online_update**.

The following appears:

Figure 4-12



From this dialog, you can view system information about the latest update and configure your YOU client.

3. To point to the local YOU server, do one of the following:
   - If you configured the YOU server as an SLP service (and completed step 1), from the Installation source drop-down menu select the ***SLP service name*** (it is normally selected by default).

     or

   - From the Installation source drop-down menu, select **User-defined location**, and in the Location field enter **http://*servername*/YOU** (where ***servername*** is the IP address or domain name of the YOU server).

You can also edit /etc/youservers and include your YOU server in that file.

4. Select or deselect the following options:
   - **Manually Select Patches.** Select this option if you want to select from a list of patches to install during the updating process.
   - **Reload All Patches From Server.** Select this option if you want all patches downloaded from the YOU server, even if they are available locally on your hard drive.
5. (Optional) Set up automatic synchronization by configuring a daily cron job:
   a. Select **Configure Fully Automatic Update**; then select **Enable Automatic Update**.

      The following appears:

Figure 4-13

   b. Enter an ***hour*** and ***minute*** when you want the updating to start each day.

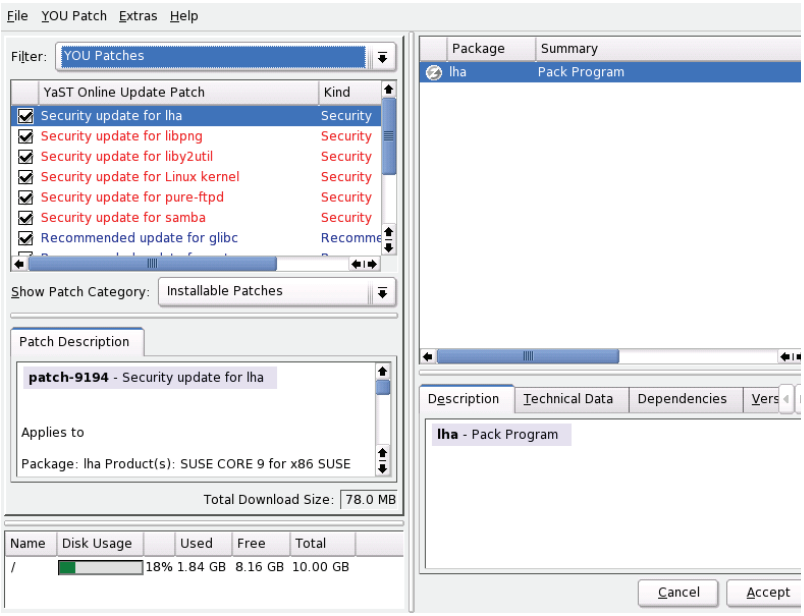
c. (Optional) If you only want to download patches, select **Only Download Patches**.

d. Select **OK**.

6. When you finish configuring, select **Next**.

   The following appears:

Figure 4-14



This dialog is similar to the YaST Install and Remove Software dialog. From this dialog you can filter the patch list view and select or deselect the patches you want to install.

7. When you finish manually selecting or deselecting patches to install, select **Accept**.

8. (Conditional) If you selected the Manually Select Patches option, select **Install Patch** or **Skip Patch** for each dialog that appears.

9. When the download and installation is complete, select **Remove Source Packages after Update**.
10. Update the system configuration by selecting **Finish**.

How to Install Patches From the Command Line

You can also run the YaST Online Update module from the command line. The following is the basic syntax:

`online_update` *`parameters`*

The following are the available parameters:

Table 4-3

Parameter	Description
-u	Base URL of the directory tree from which the patches should be fetched.
-g	Download the patches without installing them.
-i	Install already fetched patches without downloading anything.
-k	Check for existing new patches.
-c	Show current configuration without further action.
-p	Product for which patches should be fetched.
-v	Product version for which patches should be fetched.
-a	Base architecture for which patches should be fetched.
-d	Fetch patches and simulate installation for test purposes. The system remains unchanged.
-n	No signature checking of the fetched files.
-s	Display list of available patches.

(continued) **Table 4-3**

Parameter	Description
-V	Print progress messages.
-D	Debug mode for experts and for troubleshooting.

Exercise 4-4 Update SLES 9 From a Local YOU Server

As system administrator for your Digital Airlines office, you have just completed installing SUSE Linux Enterprise Server (SLES) 9 on a staging server in your lab.

As a post-installation procedure, you want to make sure you have updated your installation with the latest patches available from Novell SUSE Linux.

The following steps are specifically designed to meet the needs of a training environment where the results of updating a server need to be controlled.

For this reason, you point to the instructor server DA1 to provide the necessary patches for updating your SLES 9 installation.

However, if you are updating SLES 9 on your own (outside of a classroom), and have a valid maintenance contract, follow the steps for installing patches from a SUSE Linux download site under "How to Install Patches From a Remote YOU Server" on 4-43.

Do the following:

1. From the GUI login screen, log in as **geeko** with a password of **N0v3ll** ((uppercase **N**, zero, lowercase **v**, **3**, and two lowercase **l**'s).
2. Close any open dialogs or windows (such as the Welcome to SUSE Linux Enterprise Server 9 dialog).
3. From the KDE desktop, select the **YaST** icon; then enter the root password of **novell** and select **OK**.

   The YaST Control Center appears.
4. Select **Software > Online Update**.

   The Welcome to YaST Online Update dialog appears.

Depending on your screen resolution, you might need to adjust the size of the dialog is view all the text.

5. From the Installation source drop-down list, select **User-Defined Location**.
6. In the Location field, enter **http://DA1/YOU**.
7. Continue by selecting **Next**.

   The YOU update dialog appears with all the patches available.

   From this dialog you can filter the patch list view and select or deselect the patches you want to install.
8. From YaST Online Update Patch list, make sure the **Optional** patches (black) are deselected.
9. Make sure all the **Security** (red) and **Recommended** (blue) patches are selected.
10. Continue by selecting **Accept**.

    One or more warning messages appear.
11. For each warning message, select **Install Patch**.

    YaST downloads and installs the patches.
12. When process is complete (or during the process), select **Remove Source Packages after Update**.
13. When the patches have been installed, update the system configuration by selecting **Finish**.
14. Reboot your SLES 9 server:
    a. Press **Ctrl+Alt+Del**; then select **Logout**.

       After rebooting, you are returned to the GUI login interface.

    b. Select **Menu > Shutdown**.

c. Select **Restart computer** and enter a password of **novell**; then select **OK**.

15. After the system reboots, log back in to the KDE desktop as **geeko** with a password of **N0v3ll**.

(End of Exercise)

Summary

Objective	Summary
1. Manage RPM Software Packages	While there are several software package formats available for Linux, the format used most commonly in SUSE Linux installations is the RPM Package Manager (RPM) format. To manage installation of RPM software packages, you learned about the following: ■ RPM Components and Features ■ RPM Basics ■ How to Manage Software Packages With rpm ■ How to Manage Software Packages With YaST

Objective	Summary
2. Verify and Update Software Library Access	In addition to checking for software package dependencies, you might also need to verify that the system is configured properly to access dynamic libraries an application uses. For example, if an application that has been installed fails to start, try starting it from a terminal window. If the application reports that a library could not be found, then you might need to verify access to the dynamic libraries. To verify the libraries needed for an application, you learned about the following: ■ Software Library Basics ■ How to View Shared Library Dependencies (ldd) ■ How to Modify the Software Library Configuration File (/etc/ld.so.conf) ■ How to Update the Library Cache (/etc/ld.so.cache)

Objective	Summary
3. Manage Software Updates With YaST Online Update Server (YOU)	You can use the YaST module YOU Server Configuration to create a local update server for your network. The YOU server makes current software updates available to all YOU clients in the network. To manage software updates with a YOU server, you learned about the following: ■ YOU Basics ■ How to Configure a Local YOU Server ■ How to Configure and Use a YOU Client

SECTION 5 Manage System Initialization

In this section you learn how the SUSE Linux system boots and how to manage that process by setting runlevels, kernel parameters, boot loader options, and other system configurations.

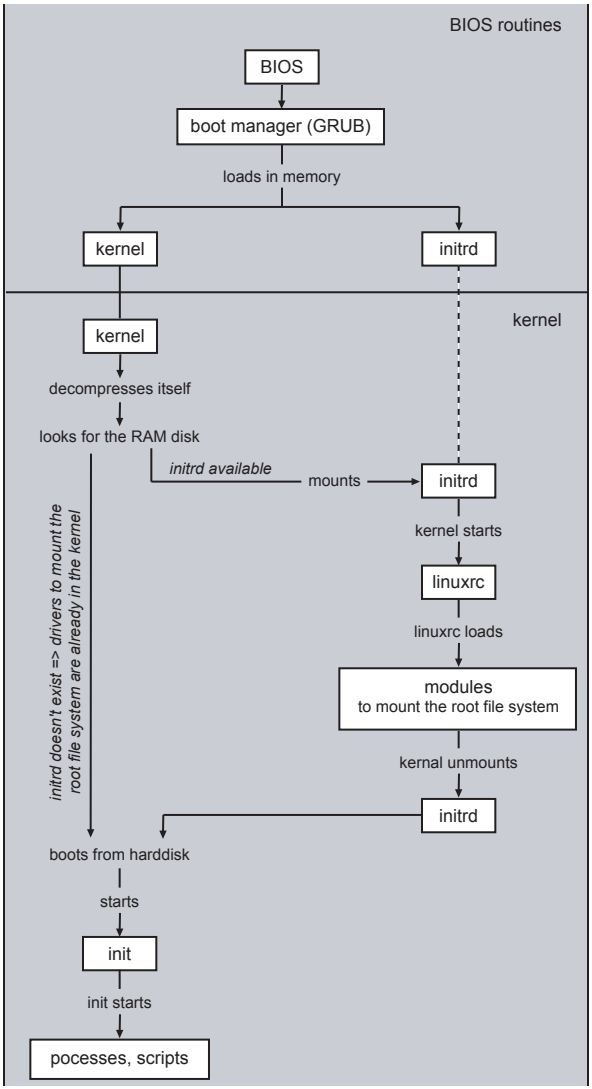
Objectives

1. Describe the Linux Load Procedure
2. Manage Runlevels
3. Manage the Kernel
4. Manage the GRUB Boot Loader
5. Modify System Settings

Objective 1 Describe the Linux Load Procedure

The following represents the basic steps of booting a computer with a Linux system installed:

Figure 5-1



The following describes the process:

- BIOS and Boot Manager
- Kernel
- initrd and linuxrc
- init

BIOS and Boot Manager

Tasks performed by the BIOS (Basic Input Output System) include performing a power-on self test, conducting the initial detection and setup of hardware, and accessing bootable devices (such as a CD or hard drive).

If the bootable device is a hard drive, BIOS also reads the MBR (master boot record). Using the code in the MBR, the BIOS starts the boot manager.

The *boot manager* (such as GRUB) loads the kernel and the initrd to memory and starts the kernel.

For details on boot managers, see "Manage the GRUB Boot Loader" on 5-42.

Kernel

The kernel uncompresses itself when you see the Uncompressing Linux... message, and then organizes and takes control of the continued booting of the system.

The kernel checks and sets the console (the BIOS registers of graphics cards and the screen output format), reads BIOS settings, and initializes basic hardware interfaces.

Next, the drivers, which are part of the kernel, probe existing hardware and initialize it accordingly.

The kernel controls the entire system, managing hardware access and allocating CPU time and memory to programs.

initrd and linuxrc

As part of the boot procedure, the kernel searches for the RAM disk, if one is available. This depends on whether the boot manager (such as GRUB) has loaded initrd.

The boot manager informs the kernel that an initrd exists and where it is located in memory. If initrd exists, it is integrated into the kernel.

If the initrd was compressed (which is typically the case), the kernel decompresses the initrd and mounts it as a temporary root file system.

A program called *linuxrc* is then started. linuxrc loads the modules required to mount the root file system.

The only requirements for the program linuxrc in the initrd are the following:

- It must have the special name linuxrc.
- It must be located in the root directory of the initrd.
- It needs to be executable by the kernel.

This means that linuxrc may be dynamically linked. In this case, the shared libraries in the directory /lib/ must be completely available in initrd.

As soon as linuxrc finishes, the initrd is unmounted and the boot process continues with the mount of the proper root file system.

If no initrd exists, the drivers to mount the file system need to be in the kernel.

What is mounted as the root file system can be influenced by linuxrc. It just needs to mount the /proc file system and write the value of the real root file system in numerical form to /proc/sys/kernel/real-rootdev.

linuxrc can also be a shell script. For this to work, a shell must exist in /bin. In other words, initrd must contain a minimal Linux system that allows the program linuxrc to be run.

When SUSE Linux is installed, a statically-linked linuxrc is used to keep initrd as small as possible. linuxrc is run with root permissions.

init

After checking the partitions and mounting the root file system, the kernel mounts from the hard drive and starts *init*, which boots the system with all its programs and configurations.

The init process is always assigned a process ID number of 1, and relies on the /etc/inittab file for configuration information on how to run the initialization process.

Once the init process starts, it begins by accessing the /etc/init.d/boot script. The /etc/init.d/boot script controls the start of services such as initializing disk quotas and mounting local file systems.

After the boot script has been completed, init starts the /etc/init.d/rc script which uses configured runlevels to start services and daemons.

Each runlevel has its own set of services that are initiated. For example, runlevel 5 includes the X Window components that run the Linux desktop.

For additional details on init, see "Runlevel Basics" on 5-7.

Objective 2 Manage Runlevels

In this objective you continue learning about the initialization process and how to manage the runlevels associated with the services initialized during the process.

To manage runlevels, you need to understand the following:

- Runlevel Basics
- How to Change the Runlevel at Boot
- How to Manage Runlevels From the Command Line
- How to Shut Down or Halt the System
- How to Set Runlevels With YaST

Runlevel Basics

To understand the basics of runlevels, you need to know the following:

- What Runlevels Are
- init Configuration File (/etc/inittab)
- init Scripts
- Runlevel Symbolic Links
- How init Determines Which Services to Start and Stop
- Activate and Deactivate Services for a Runlevel

What Runlevels Are

In Linux, various runlevels define the state of the system. Which runlevel the system starts in when it is booted is defined in the file /etc/inittab by the entry initdefault.

The init runlevels available on SUSE Linux Enterprise Server 9 include:

Table 5-1

Runlevel	Action
0	Shutdown Linux system
1	Single user text mode
2	Local multi-user without remote network (such as NFS)
3	Full multi-user text mode (with networking)
4	Not used (user-definable)
5	Full multi-user graphical mode (with an X Windows login dialog, networking, and a desktop)
6	Reboot Linux system

You can find out the runlevel of your current environment at a command line by entering **runlevel**.

init Configuration File (/etc/inittab)

As mentioned previously, initialization of the system is done by /sbin/init, started by the kernel as the first process of the system.

The init process (or one of its child processes) starts all additional processes. This means that init controls the entire booting of the system. And because init is the last process running, it also controls the shutting down of the system, ensuring that all other processes correctly end.

Because of this position of priority, signal 9 (SIGKILL), which you can normally use to end all processes, has no effect on init.

The basic configuration file of init is /etc/inittab. This file determines what happens on individual runlevels. Various scripts are started by init, depending on these entries. All these scripts are located in the directory /etc/init.d/.

To understand the contents of the file /etc/inittab, you need to know the following:

- inittab Syntax
- inittab Standard Entries

inittab Syntax

The following is the syntax of each line in the file /etc/inittab:

```
id:rl:action:process
```

The following describes the parameters:

- ***id.*** A unique name for the entry in /etc/inittab. It can be up to four characters long.
- ***rl.*** Refers to one or more runlevels in which this entry should be evaluated.
- ***action.*** Describes what init is to do.
- ***process.*** Is the process connected to this entry.

inittab Standard Entries

The first entry in the file /etc/inittab contains the following parameters:

```
id:5:initdefault:
```

The parameter initdefault signals to the init process which level it should bring the system to. This can be overwritten at the boot prompt by entering a different level.

The following is the next entry:

```
si:bootwait:/etc/init.d/boot
```

The parameter bootwait indicates to carry out this command while booting and wait until it has finished.

The next few entries describe the actions for runlevels 0 to 6:

```
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
#l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

The parameter wait means that when the system changes to the indicated level, the appropriate command is carried out and init waits until it has been completed. The parameter also means that further entries for the level are only performed after this process is completed.

The single user mode is a special case:

```
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin
```

First, the command to initialize the level is performed. Runlevel S is used by the scripts that are run when changing to runlevel 1.

Then the command sulogin is started. sulogin is intended only for the system administrator to log in.

The parameter respawn indicates to init to wait for the end of the process then restart it.

For those accustomed to PCs, /etc/inittab also defines the Ctrl+Alt+Del key combination for restarting:

```
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

The action ctrlaltdel is carried out by the init process only if these keys are pressed. If you do not want to allow this action, comment out (#) or remove the line.

The final large block of entries describe in which runlevels getty processes (login processes) are started:

```
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

The getty processes provide the login prompt and in return expect a user name as input. They are started in runlevels 2, 3, and 5.

Runlevel 4 in the above example is ignored because the line that defines the actions for the runlevel is commented out earlier in the file (**#l4:4:wait:/etc/init.d/rc 4**).

If a session ends, the processes are started again by init. If a line is disabled here, no further login is possible at the corresponding virtual console.

If /etc/inittab is damaged, the system might not boot properly. For this reason, you need to be extremely careful while editing /etc/inittab and always keep a backup of an intact version.

To repair damage, try entering **init=/bin/bash** after the kernel name at the boot prompt to boot directly into a shell. After that, replace /etc/inittab with your backup version using the command cp.

init Scripts

All the scripts used by init to start and stop services are located in the directory /etc/init.d/, as in the following:

Figure 5-2

```
DA50:/etc/init.d # ls -l
total 955
drwxr-xr-x  11 root root  4896 Jul  9 08:05 .
drwxr-xr-x  92 root root  8736 Aug  6 13:05 ..
-rw-r--r--   1 root root  7000 Jun 30 12:37 README
-rwxr-xr-x   1 root root  1640 Jun 30 12:24 SuSEfirewall2_final
-rwxr-xr-x   1 root root  1628 Jun 30 12:24 SuSEfirewall2_init
-rwxr-xr-x   1 root root  1856 Jun 30 12:24 SuSEfirewall2_setup
-rwxr-xr-x   1 root root  3702 Jun 30 12:07 acct
-rwxr--r--   1 root root  4153 Jun 30 12:23 acpid
-rwxr-xr-x   1 root root  2151 Jun 30 13:04 adsl
-rwxr-xr-x   1 root root  6537 Jun 30 16:26 alsasound
-rwxr-xr-x   1 root root  2391 Jun 30 16:40 amavis
-rwxr--r--   1 root root  8630 Jun 30 16:40 apache2
-rwxr-xr-x   1 root root  3638 Jun 30 15:17 argus
-rwxr-xr-x   1 root root  3849 Jun 30 16:22 arpwatch
-rwxr-xr-x   1 root root  2792 Jun 30 13:30 atalk
-rwxr-xr-x   1 root root  3689 Jun 30 16:23 atd
-rwxr--r--   1 root root  8946 Jun 30 16:10 autofs
-rwxr-xr-x   1 root root  2954 May 26 13:22 autoyast
-rwxr-xr-x   1 root root  2694 Jun 30 12:08 avgate
-rwxr-xr-x   1 root root  3523 Jun 30 13:43 bgpd
-rwxr-xr-x   1 root root  7096 Mar 26 07:07 boot
-rwxr-xr-x   1 root root  2494 Mar  8 03:44 boot.clock
```

These scripts can be called up in the following ways:

- Directly by init when you boot the system, when the system is shut down, when you stop the system with **Ctrl+Alt+Del**, or when there is a power failure.
- Indirectly by init when you change the runlevel. In this case, it is the script /etc/init.d/rc that runs the necessary scripts in the correct order during the runlevel change.
- Directly by the /etc/init.d/ script start or stop commands.

  You can also enter **rcscript start** or **stop** if corresponding links are set in /sbin/ or /usr/sbin/.

Each of the scripts in /etc/init.d is run both as a start script and a stop script. For this reason, they must understand the following parameters:

Table 5-2

Parameter	Description
start	Starts a service that is not running.
restart	Stops a running service and restarts it.
stop	Stops a running service.
reload	Rereads the configuration of the service without stopping and restarting the service itself.
force-reload	Reloads the configuration if the service supports this. Otherwise, it does the same thing as restart.
status	Displays the current status of the service.

The following describes some of the more important scripts stored in /etc/init.d:

- **boot.** This script is started directly by init when the system starts. It is run once and once only. It evaluates the directory /etc/init.d/boot.d/ and starts all the scripts linked to filenames with an "S" at the beginning of their names (see "Runlevel Symbolic Links" on 5-14).

  These scripts perform the following tasks:

  - Starts the kernel daemon, which takes over the automatic loading of kernel modules
  - Checks the file systems
  - Deletes unnecessary files in /var/lock/
  - Sets the system time
  - Configures PnP hardware with the isapnp tools

System extensions are activated from the script /etc/init.d/boot.local (you can add your own system extensions to this script)

- **boot.local.** This script includes additional commands to execute at boot before changing into a runlevel. It can be compared to AUTOEXEC.BAT on a DOS systems.
- **boot.setup.** This script is run when changing from single user mode to any other runlevel and is responsible for a number of basic settings, such the keyboard layout and initialization of the virtual consoles.
- **halt.** This script is run if runlevel 0 or 6 is started. It is called up either with the command halt (the system is completely shut down) or with the command reboot (the system is shut down and then rebooted).
- **rc.** This script is responsible for the correct change from one runlevel to another. It runs the stop scripts for the current runlevel, and then it runs the start scripts for the new one.

To create your own scripts, you can use the file /etc/init.d/skeleton as a template.

Runlevel Symbolic Links

For each runlevel, there is a corresponding subdirectory in /etc/init.d/. For runlevel 1 it is /etc/init.d/rc1.d/, for runlevel 2 it is /etc/init.d/rc2.d/, and so on.

When you view the files in a directory such as /etc/init.d/rc3.d/, you see two kinds of files—those that start with a “K” and those that start with an “S”:

Figure 5-3

K10sshd	3.6 KB	Shell Script	2003-03-18 06:15	rwxr-xr-x	root
K12nfs	2.5 KB	Shell Script	2004-06-21 09:10	rwxr-xr-x	root
K12nfsboot	1.1 KB	Shell Script	2004-06-30 12:59	rwx------	root
K14portmap	3.5 KB	Shell Script	2004-06-30 12:49	rwxr--r--	root
K14resmgr	3.4 KB	Shell Script	2004-06-30 11:56	rwxr-xr-x	root
K14smbfs	4.5 KB	Shell Script	2004-06-01 08:53	rwxr-xr-x	root
K14splash_early	617 B	Shell Script	2004-06-30 12:45	rwxr-xr-x	root
K16syslog	2.8 KB	Shell Script	2004-06-30 12:24	rwxr-xr--	root
K17network	15.9 KB	Shell Script	2004-07-01 06:36	rwxr-xr-x	root
K20coldplug	2.9 KB	Shell Script	2004-06-30 12:03	rwxr-xr-x	root
K21hotplug	2.0 KB	Shell Script	2004-06-30 12:03	rwxr-xr-x	root
K21isdn	5.4 KB	Shell Script	2004-05-03 10:24	rwxr-xr-x	root
K21random	1.7 KB	Shell Script	2003-09-01 05:11	rwxr-xr-x	root
S01hotplug	2.0 KB	Shell Script	2004-06-30 12:03	rwxr-xr-x	root
S01isdn	5.4 KB	Shell Script	2004-05-03 10:24	rwxr-xr-x	root
S01random	1.7 KB	Shell Script	2003-09-01 05:11	rwxr-xr-x	root
S02coldplug	2.9 KB	Shell Script	2004-06-30 12:03	rwxr-xr-x	root
S05network	15.9 KB	Shell Script	2004-07-01 06:36	rwxr-xr-x	root
S06syslog	2.8 KB	Shell Script	2004-06-30 12:24	rwxr-xr--	root
S08portmap	3.5 KB	Shell Script	2004-06-30 12:49	rwxr--r--	root
S08resmgr	3.4 KB	Shell Script	2004-06-30 11:56	rwxr-xr-x	root
S08smbfs	4.5 KB	Shell Script	2004-06-01 08:53	rwxr-xr-x	root
S08splash_early	617 B	Shell Script	2004-06-30 12:45	rwxr-xr-x	root

The first letter is always followed by 2 digits and the name of a service. Whether a service is started in a specific runlevel depends on whether there are **S*xxservice*** and **K*xxservice*** files in the **/etc/init.d/rc*x*.d/** directory.

Entering **ls -l** in an /etc/init.d/rcx.d/ directory indicates that these files are actually symbolic links pointing to service scripts in /etc/ini.d/ (as in the following):

Figure 5-4

```
DA50:~ # cd /etc/init.d/rc3.d
DA50:/etc/init.d/rc3.d # ls -l
total 6
drwxr-xr-x   2 root root 1864 Aug  4 13:35 .
drwxr-xr-x  11 root root 4896 Jul  9 08:05 ..
lrwxrwxrwx   1 root root   14 Jul  9 07:59 K02splash_late -> ../splash_
late
lrwxrwxrwx   1 root root    7 Jul  9 07:59 K03cron -> ../cron
lrwxrwxrwx   1 root root    9 Jul 20 18:27 K03xinetd -> ../xinetd
lrwxrwxrwx   1 root root   10 Aug  4 13:35 K04apache2 -> ../apache2
lrwxrwxrwx   1 root root   12 Jul 21 13:00 K05nfsserver -> ../nfsserver
lrwxrwxrwx   1 root root   10 Jul  9 07:59 K05postfix -> ../postfix
lrwxrwxrwx   1 root root    7 Jul  9 07:59 K06nscd -> ../nscd
lrwxrwxrwx   1 root root    7 Jul  9 07:59 K07cups -> ../cups
lrwxrwxrwx   1 root root    9 Jul  9 07:59 K08hwscan -> ../hwscan
lrwxrwxrwx   1 root root    7 Jul  9 07:59 K08slpd -> ../slpd
lrwxrwxrwx   1 root root   13 Jul  9 07:51 K09powersaved -> ../powersav
ed
lrwxrwxrwx   1 root root    9 Jul  9 07:51 K09splash -> ../splash
lrwxrwxrwx   1 root root   12 Jul  9 08:26 K10alsasound -> ../alsasound
lrwxrwxrwx   1 root root    8 Jul  9 07:51 K10fbset -> ../fbset
lrwxrwxrwx   1 root root   14 Aug  3 16:45 K10powertweakd -> ../powertw
eakd
lrwxrwxrwx   1 root root   17 Jul  9 07:58 K10running-kernel -> ../runn
```

Some services point to the same script. For example, if you enter **ls -l *network** in the /etc/init.d/rc3.d/ directory, you see that 2 network services both point to the script /etc/init.d/network:

Figure 5-5

```
DA50:/etc/init.d/rc3.d # ls -l *network
lrwxrwxrwx  1 root root 10 Jul  9 08:20 K17network -> ../network
lrwxrwxrwx  1 root root 10 Jul  9 08:20 S05network -> ../network
DA50:/etc/init.d/rc3.d #
```

By using symbolic links in subdirectories, only the script in /etc/init.d/ needs to be modified.

Sometimes K*xx* links are referred to as *kill scripts*, while S*xx* links are referred to as *start scripts*. In fact there are no separate scripts for starting and stopping services—just the parameters stop and start.

How init Determines Which Services to Start and Stop

You already know that a service is started with the parameter start, and stopped with the parameter stop. The same parameters are also used when changing from one runlevel to another.

The script /etc/init.d/rc examines the directories /etc/init.d/rccurrentrl.d/ and /etc/init.d/rcnewrl.d/ and determines what to do. The following are 3 possibilities:

- There is a **K*xx*** link for a certain service in /etc/init.d/rccurrentrl.d/ and there is an **S*xx*** link in /etc/init.d/rcnewrl.d/ for the same service.

  In this case, the service is neither started nor stopped; the corresponding script in /etc/init.d/ is not called at all.

- There is a **K*xx*** link for a certain service in /etc/init.d/rccurrentrl.d/ and there is no corresponding **S*xx*** link in /etc/init.d/rcnewrl.d/.

  In this case, the script in /etc/init.d/service is called with the parameter stop and the service is stopped.

- There is an **S*xx*** link in /etc/init.d/rcnewrl.d/ and there is no corresponding **K*xx*** link for the service in /etc/init.d/rccurrentrl.d/.

  In this case, the script in /etc/init.d/service is called with the parameter start and the service is started.

The number after the K or S determines the sequence in which the scripts are called.

For example, script K10serviceA is called before script K20serviceB, which means that serviceA is shut down before serviceB.

Script S15serviceC is called before S23serviceD, which means that serviceC starts before serviceD. This is important if serviceD depends on a running serviceC.

For example the following happens when you change from runlevel 3 to runlevel 5:

1. You tell init to change to a different runlevel by entering (as root) **init 5**.
2. init checks its configuration file (/etc/inittab) and determines it should start /etc/init.d/rc with the new runlevel (**5**) as a parameter.
3. rc calls the stop scripts (**K*xx***) of the current runlevel for those services for which there is no start script (**S*xx***) in the new runlevel.
4. The start scripts in the new runlevel for those services for which there was no kill script in the old runlevel are launched.

When changing to the same runlevel as the current runlevel, init only checks /etc/inittab for changes and starts the appropriate steps (such as starting a getty on another interface).

Activate and Deactivate Services for a Runlevel

The services in a runlevel can be activated and deactivated from the command line with the command insserv or by using YaST.

Although you could create symbolic create links in the runlevel subdirectories yourself to modify services that are stopped and started, an easier way is to edit the header of a script.

The INIT INFO block at the beginning of the script determines in which runlevel the service that the script controls should start or stop:

```
### BEGIN INIT INFO
# Provides: syslog
# Required-Start: network
# Required-Stop: network
# Default-Start: 2 3 5
# Default-Stop:
# Description: Start the system logging daemons
### END INIT INFO
```

The INIT INFO block is used by the program insserv to determine in which runlevel subdirectories links need to be placed and what numbers need to be put after K and S.

For details on the program insserv, enter **man 8 insserv**.

The entry Default-Start determines in which runlevel directories links are to be placed. The entry Required-Start determines which services have to be started before the one being considered.

After editing the INIT INFO block, enter **insserv** to create the needed links and renumber the existing ones as needed.

To remove all links for a service (disabling the service), stop the service (if it is running) by entering **/etc/init.d/*service* stop**, and then enter **insserv -r *service***.

You can also use the YaST runlevel editor to set these links. We recommend that you choose one method or the other. Switching between methods can lead to errors.

How to Change the Runlevel at Boot

The standard runlevel is 3 or 5. However, you can boot to another runlevel. By default, at system start-up GRUB offers the following 3 choices:

- Linux
- Floppy
- Failsafe

When you select one of these entries, additional options are displayed in the field boot options (you might need to press additional keystrokes to display the options).

For the entry Linux, this is the option starting root=/dev/hd, which tells the kernel the location of the root partition of the system.

In addition, the option vga=, with the resolution for the framebuffer device, is specified in most cases.

At this point, you can indicate the runlevel at which you want the system to start. This parameter is passed to init.

The following is an example entry at boot options:

```
root=/dev/hda4 vga=791 1
```

As root partition, /dev/hda4 is transmitted to the kernel. The framebuffer is configured, and the system boots to runlevel 1 (single user mode for administration) with a resolution for the framebuffer set at 791.

How to Manage Runlevels From the Command Line

You can change to another runlevel after the system is running by using the command init. For example, you can change to runlevel 1 from a command line by entering **init 1**.

In the same way, you can change back to the standard runlevel where all programs needed for operation are run and where individual users can log in to the system.

For example, you can return to a full GUI desktop and network interface (runlevel 5) by entering **init 5**.

If the partition /usr of a system is mounted through NFS, you should not use runlevel 2 because NFS file systems are not available in this runlevel.

You stop the system by entering **init 0**; you restart the system by entering **init 6**.

Runlevels are useful if you encounter problems caused by a particular service (X or network) in a higher runlevel. In this case, you can switch the system to a lower runlevel to repair the service.

Many servers operate without a graphical user interface and must be booted in a runlevel without X windows (such as runlevel 3).

If the graphical user interface freezes at any time, press **Ctrl+Alt+Backspace** to restart the X window system.

You can also restart the X window system by switching to a text console with **Ctrl+Alt+F1**, logging in as root, and switching to runlevel 3 with the command **init 3**.

This shuts down your X window system, leaving you with a text console. To restart the graphical system, enter **init 5**.

Sometimes a remote login is required to enter **init 3**.

How to Shut Down or Halt the System

Like most modern operating systems, Linux reacts sensitively to being switched off without warning. If this happens, the file systems need to be checked and corrected before the system can be used again.

For this reason, the system should always be shut down properly. With the appropriate hardware, Linux can also switch off the machine as the last stage of shutting down.

Although you can halt the system by changing to runlevel 0 and restarting in runlevel 6, the following are some other useful commands for properly shutting down the system or restarting it:

Table 5-3

Command	Description
halt	This command ensures an immediate, controlled system halt. All processes are stopped and the system no longer reacts to any input. You can now switch off the computer, if it is not configured to switch off automatically.
poweroff	This command has the same effect as halt, except that the machine is switched off automatically (if the hardware allows it).
reboot	This command reboots the system.

(continued) **Table 5-3**

Command	Description
shutdown -h *time*	This command shuts down the system after the specified *time*: +m (number of minutes from now), hh:mm (time in hours:minutes, when Linux should shut down), and now (system is stopped immediately). If you use the option -r instead of -h, the system is rebooted (runlevel 6). Without options, it changes to runlevel 1 (single user mode).

The command shutdown controls the shutdown of the system in a special way, compared with the other stop commands. The command informs all users that the system will be shut down and does not allow other users to log in before it shuts down.

The command shutdown can also be supplied with a warning message, such as the following:

```
shutdown +5 The new hard drive has arrived
```

If a shutdown planned for a later time should not be carried out after all, you can revoke the shutdown by entering **shutdown -c**.

How to Set Runlevels With YaST

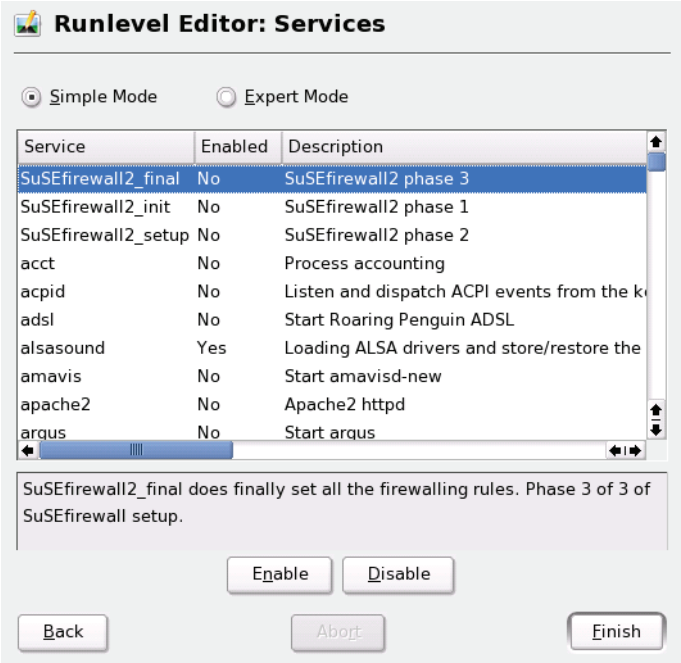
To set runlevels with YaST, do the following:

1. From the KDE desktop, start the YaST Runlevel Editor module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > Runlevel Editor**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 runlevel**.

The following appears:

Figure 5-6



From this dialog, you can select from the following modes:

- **Simple Mode.** This mode displays a list of all available services and the current status of each service.

  You can select a service, and then select **Enable** or **Disable**.

  Selecting Enable starts the service (and services it depends on) and enables them to start at system boot time. Selecting Disable stops dependent services and the service itself and disables their start at system boot time.

- **Expert Mode.** This mode gives you control over the runlevels in which a service is started or stopped, and lets you change the default runlevel.

2. Switch to the Expert mode by selecting **Expert Mode**.

   The following appears:

Figure 5-7

   In this mode, the dialog displays the current default runlevel at the top.

3. (Optional) From the default runlevel drop-down list, select a new ***default runlevel***.

   Normally, the default runlevel of a SUSE Linux system is runlevel 5 (full multiuser with network and xdm). A suitable alternative might be runlevel 3 (full multiuser with network).

   Changes to the default runlevel take effect the next time you boot your computer

4. From the list of services, select a **service**; then from the options below the list, select the ***runlevels*** you want associated with the service.

   The list includes the services and daemons available, indicates whether they are currently enabled on your system, and lists the runlevels currently assigned.

   Runlevel 4 is initially undefined to allow creation of a custom runlevel.

5. (Optional) If you want a service activated after editing the runlevels, from the drop-down list select **Start now**, **Stop now**, or **Refresh status**.

   You can use Refresh status to check the current status (if this has not been done automatically).

6. From the Set/Reset drop-down list, select one of the following:
   - **Enable the service:** activates the service in the standard runlevels.
   - **Disable the service:** deactivates the service.
   - **Enable all services:** Activates all services in their standard runlevels.

7. When you finish configuring the runlevels, save the configuration by selecting **Finish**.

Remember that faulty runlevel settings can make a system unusable. Before applying your changes, make absolutely sure you know about the impact of the changes.

Exercise 5-1 Manage Run Levels

Do the following:

1. From the KDE desktop, open a terminal window; then su to root (**su -**) with a password of **novell**.
2. Check the previous and current runlevels by entering **runlevel**.

   List the runlevels:

Table 5-4

Previous	Current

   Notice that the previous runlevel is listed as N, which means that there was no previous runlevel set.

3. Change to runlevel 3 by entering **telinit 3**.

   The KDE desktop (X windows) is terminated and you are left at a terminal login prompt.

4. Log in as **root** with a password of **novell**.
5. Check the previous and current runlevel by entering **runlevel**.

   List the runlevels:

Table 5-5

Previous	Current

6. Switch to runlevel 5 by entering **init 5**.

   The GUI login screen appears.

7. Log in as **geeko** with a password of **N0v3ll**.

(End of Exercise)

Objective 3 Manage the Kernel

The primary function of the Linux kernel is to manage the system hardware resources, making them available to various system processes.

To manage the kernel, you need to understand the following:

- Kernel Module Basics
- How to Find Hardware Driver Information
- How to Manage Modules From the Command Line
- modprobe Configuration File (/etc/modprobe.conf)
- Kernel Module Loader (kmod)

For the latest kernel documentation, see /usr/src/linux/Documentation.

Kernel Module Basics

The kernel that is installed in the directory /boot/ is configured for a wide range of hardware. It is not necessary to compile a custom kernel, unless you want to test experimental features and drivers.

Drivers and features of the Linux kernel can either be compiled into the kernel or be loaded as kernel modules. These modules can be loaded later, while the system is running, without having to reboot the computer.

This is especially true of kernel modules that are not required to boot the system. By loading them as components after the system boots, the kernel can be kept relatively small.

The kernel modules are located in the directory /lib/modules/*version*/kernel/.

For example, the modules for the 2.6 kernel can be found in the following directory:

/lib/modules/2.6.5-7.79-default/kernel/

The following are files and directories related to the kernel:

- **/boot/initrd.** A link to the initrd module.
- **/boot/vmlinuz.** A link to the kernel of the SUSE Linux Enterprise Server.
- **/proc/sys/kernel/.** The directory with information about the kernel.
- **/proc/version.** Features the version of the current kernel.
- **/usr/src/linux/.** The directory containing the source files of the kernel.

How to Find Hardware Driver Information

You can use the command *hwinfo* to detect the hardware of your system and select the drivers needed to run this hardware.

For a short introduction to this command, enter **hwinfo --help**. For specific hardware information, enter **hwinfo --*hardware_type*** (such as **hwinfo --scsi**).

All this information is also available in YaST in the Hardware Information module (**Hardware > Hardware Information**).

Exercise 5-2 View Information About the Hardware System

In this exercise, you do the following:

- Part I: View General Information About the Hardware System
- Part II: View Information About Specific Hardware

Part I: View General Information About the Hardware System

Do the following:

1. View hardware information from the command line:
   a. Open a terminal window and su to root (**su -**) with a password of **novell**.
   b. View the devices used on your Linux system listed in /proc/devices by entering **less /proc/devices**.
   c. Return to the command line by typing **q**.
   d. View the system devices on your SLES 9 server by entering **hwinfo | less**.

      It takes a few moments for the information to be gathered and displayed.
   e. Scroll through the information by pressing the **Spacebar**.
   f. Exit the information screen by typing **q**.
2. View and save hardware information from YaST:
   a. From the KDE desktop, select the **YaST** icon; then enter a password of **novell** and select **OK**.

      The YaST Control Center appears.
   b. Select **Hardware > Hardware Information**.

      After a few moments of gathering information, the Hardware info dialog appears.
   c. Try expanding several entries (such as **Display** or **Network Card**) to view information about your server.

d. Save the hardware information by selecting **Save to File**; then save the information as **server_hwinfo** in the directory **/files/private**.

e. When you finish, close the Hardware info dialog by selecting **Close**; then close the YaST Control Center.

f. Open the File Manager in super user mode (root) by pressing **Alt+F2** and entering **kdesu konqueror**; then select **Run**.

g. Enter a password of **novell**; then select **OK**.

h. In the Location field, view the file server_hwinfo by entering **/files/private/server_hwinfo**.

i. Scroll through the file and note the available information.

j. Close the Konqueror window.

3. View and configure hardware from suseplugger:

a. From the KDE desktop, view hardware information using suseplugger by selecting the **SUSE Hardware Tool** icon in the system tray (at the bottom right of the screen).

The suseplugger dialog appears.

b. Expand the **Network Controller** hardware category.

c. Select a ***network card*** under the category.

Notice that the Configure and Details buttons become active.

d. Select **Configure**; then enter a password of **novell** and select **OK**.

The YaST Network cards configuration dialog appears.

By using suseplugger, you can directly access the YaST module you need to configure the hardware you selected.

e. Close the dialog by selecting **Abort**; then select **Yes**.

f. From the suseplugger dialog, select **Details**.

A dialog appears with 3 tabs: General, Resources, and Driver.

g. Select each tab to view the information.

h. When you finish, close the dialog by selecting **OK**.

i. Close the suseplugger dialog by selecting **Close**.

You can also start suseplugger from a command line by logging in as root (**sux -**), and then entering **suseplugger**.

Part II: View Information About Specific Hardware

Do the following:

1. View CPU information from the command line:
   a. From the terminal window, change to the directory /proc/ by entering **cd /proc**.
   b. View the contents of the file /proc/cpuinfo by entering **cat cpuinfo**.
   c. Monitor only CPU utilization at 5 second intervals 10 times by entering **iostat -c -t 5 10**.
   d. When you finish viewing the information, exit the display by pressing **Ctl+C**.
   e. View CPU utilization every second by entering **top -d 1**.

      Note the headings at the top of the list (primarily the %CPU column).
   f. Try sorting the processes by % Memory (type **F**, then enter **n**), by % CPU (type **F**, then enter **k**), and by user (type **F**, then enter **e**).
   g. When you finish, exit the top display by typing **q**.
   h. View the CPU utilization for each process by entering **ps -aux | less**.
   i. Page through the processes by pressing the **Spacebar**; then exit the display by typing **q**.
2. From the KDE menu, view current CPU statistics from Xosview by selecting **System > Monitor > X osview**.

Close the display when you finish monitoring the CPU.

3. View CPU information from KDE System Guard:

   a. From the KDE Menu, select **System > Monitor > KDE System Guard**.

      The KDE System Guard dialog appears with the System Load tab selected.

      From this tab you can view updated information about the CPU load, load average (1 minute), physical memory, and swap memory.

      Besides the preconfigured tabs, you can also create your own worksheet of specific system information.

   b. Select **File > New**.

      A Worksheet Properties dialog appears.

   c. Enter the following properties:

      - Title: **CPU Information**
      - Rows: **2**
      - Columns: **2**

   d. Continue by selecting **OK**.

      A CPU Information tab appears with 4 areas for dropping sensors.

   e. From the Sensor Browser panel (on the left) expand **localhost > CPU Load**.

   f. Select and drag the following to the 4 sensor areas (with the indicated display styles):

      - **Idle Load** (Signal Plotter)
      - **Load Average (1 minute)** (Multimeter)
      - **System Load** (Bar Graph)
      - **User Load** (Signal Plotter)

   g. Maximize the KDE System Guard window.

h. (Optional) Right-click a sensor area; then select **Properties** and change the properties of the display (such as the grid background color).

i. When you finish using KDE System Guard, select **File > Quit**; then select **No**.

4. View device information using /proc:

a. Make sure you are su'd to root from a terminal window.

b. From the command line, view the I/O ports used on your Linux system by entering **cat /proc/ioports**.

c. View the IRQs used on your Linux system by entering **cat /proc/interrupts**.

d. View the DMA channels used on your Linux system by entering **cat /proc/dma**.

e. View the PCI devices used on your Linux system by entering **cat /proc/bus/pci/devices | less** or by entering **lspci**.

5. View hard disk information:

a. View the geometry of the first IDE disk on your Linux system by entering **fdisk -l /dev/hda**.

Using fdisk you can see the number of cylinders on your disk.

b. View the hard disk settings by entering **hdparm -a /dev/hda**.

6. View SCSI information:

You can find SCSI information by checking the files in /proc/scsi/, or a information summary in /proc/scsi/scsi.

a. View the SCSI summary information by entering **cat /proc/scsi/scsi**.

You can find detailed information about the configuration in the files in /proc/scsi/sg/.

b. Change to the directory /proc/scsi/sg/ by entering **cd /proc/scsi/sg**.

c. List the files by entering **ls -l**.

d. View one or more files in the directory by entering **cat *filename*** (such as **cat version**).

e. Run a scan of the SCSI bus by entering **sg_scan**.

f. Check for SCSI devices on your system by entering **scsiinfo -l**.

7. View ISA devices by entering **pnpdump | less**.

8. View USB information:

a. View the USB devices by entering **lsusb**.

b. View detailed information about USB devices and drivers from the proc file system by entering the following commands:

cat /proc/bus/usb/devices
cat /proc/bus/usb/drivers

If there are no USB drivers on your server, the directory /proc/bus/usb/drivers does not exist.

c. Determine the kernel modules available for the USB plugged-in device in /proc/bus/usb/001/001 by entering the following:

usbmodules --device /proc/bus/usb/001/001

9. Close the terminal window.

(End of Exercise)

How to Manage Modules From the Command Line

The following are commands you can use from a command line when working with modules:

- **lsmod.** This command lists the currently loaded modules in the kernel.

  The following is an example:

```
DA50:~ # lsmod
Module                  Size  Used by
quota_v2               12928  2
edd                    13720  0
joydev                 14528  0
sg                     41632  0
st                     44956  0
sr_mod                 21028  0
ide_cd                 42628  0
cdrom                  42780  2 sr_mod,ide_cd
nvram                  13448  0
usbserial              35952  0
parport_pc             41024  1
lp                     15364  0
parport                44232  2 parport_pc,lp
ipv6                  276348  44
uhci_hcd               35728  0
intel_agp              22812  1
agpgart                36140  1 intel_agp
evdev                  13952  0
usbcore               116572  4 usbserial,uhci_hcd
```

  The list includes information about the module name, size of the module, how often the module is used, and by which other modules use it.

- **insmod *module*.** This command loads the indicated *module* into the kernel.

  The module must be stored in the directory /lib/modules/*version_number*/. However, it is recommended to use modprobe for loading modules.

- **rmmod *module*.** This command removes the indicated *module* from the kernel. However, it can only be removed if no processes are accessing hardware connected to it or corresponding services.

  We recommend that you use **modprobe -r** for removing modules.

- **modprobe *module*.** This command loads the indicated *module* into the kernel or removes it (with option -r).

  Dependencies of other modules are taken into account when using modprobe. In addition, modprobe reads in the file /etc/modprobe.conf for any configuration settings.

  This command can only be used if the file /lib/modules/*version*/modules.dep created by the command depmod exists. This file is used to add or remove dependencies.

  The kernel daemon (Kmod since kernel version 2.2.x) ensures that modules needed in the running operation are automatically loaded using modprobe (such as accessing the CD-ROM drive).

For more detailed information, enter **man modprobe**.

- **depmod.** This command creates the file /lib/modules/*version*/modules.dep. This file contains the dependencies of individual modules on each other.

  When a module is loaded (such as with modprobe), modules.dep ensures that all modules it depends on are also loaded.

  If the file modules.dep does not exist, it is created automatically when the system starts by the start script /etc/init.d/boot. For this reason, you do not need to create the file manually.

- **modinfo *option module*.** This command displays information (such as license, author, and description) about the module indicated on the command line.

The following is an example:

```
DA50:~ # modinfo isdn
license:        GPL
author:         Fritz Elfert
description:    ISDN4Linux: link layer
depends:        slhc
supported:      yes
vermagic:       2.6.5-7.21-default 586 REGPARM gcc-3.3
```

For more detailed information, enter **man modinfo**.

modprobe Configuration File (/etc/modprobe.conf)

The file /etc/modprob.conf is the configuration file for the kernel modules. For example, it contains parameters for the modules that access hardware directly.

The file plays an important role in loading modules. Various command types can be found in the file, such as the following:

- **install.** These instructions let modprobe execute commands when loading a specific module into the kernel.

  The following is an example:

  ```
  install     eth0     /bin/true
  ```

- **alias.** These instructions determine which kernel module will be loaded for a specific device file.

  The following is an example:

  ```
  alias     eth0     nvnet
  ```

- **options.** These instructions are options for loading a module.

  The following is an example:

  ```
  options     ne     io=0x300 irq=5
  ```

For more detailed information, enter **man 5 modprobe.conf**.

Kernel Module Loader (kmod)

The kernel module loader (kmod) is the most elegant way to use modules. Kmod performs background monitoring and makes sure the required modules are loaded by modprobe as soon as the respective functionality is needed in the kernel.

To activate kmod, the option Kernel module loader (CONFIG_KMOD) needs to be set to “y” (yes) in the kernel configuration. This is the default setting for SLES 9.

Kmod is not designed to unload modules automatically; compared with today’s RAM capacities, the potential memory savings would be marginal.

For performance reasons, monolithic kernels might be more suitable for servers that are used for special tasks and need only a few drivers.

Exercise 5-3 Manage the Linux Kernel

In this exercise, you view information about your kernel and load and unload kernel modules.

Do the following:

1. From a terminal window, su to root (**su -**) with a password of **novell**.
2. View the currently loaded kernel modules by entering **lsmod**.
3. Scroll through the modules to see if the joystick module (**joydev**) is loaded.

   The 0 in the Used column indicates that the module is not in use.
4. Remove the joystick module from the kernel memory by entering **rmmod joydev**.
5. Verify that the joydev kernel module was removed from memory by entering **lsmod**.

   Notice that the module joydev is no longer listed.
6. Load the joystick kernel module by entering **modprobe joydev**.
7. Verify that the joydev kernel module is loaded in memory by entering **lsmod**.
8. View the kernel modules configuration by entering the following:

   modprobe -c | less
9. Scroll through the configuration information by pressing the **Spacebar**.
10. When you finish, return to the command line by typing **q**.
11. Create a list of kernel modules dependencies by entering **depmod -v | less**.

    It takes a few moments for the information to be generated.

12. Scroll through the dependency information by pressing the **Spacebar**.
13. When you finish, return to the command line by typing **q**.
14. Close the terminal window by entering **exit** twice.

(End of Exercise)

Objective 4 Manage the GRUB Boot Loader

To manage the GRUB boot loader, you need to know the following:

- What a Boot Manager Is
- Boot Managers in SUSE LINUX
- How to Start the GRUB Shell
- How to Modify the GRUB Configuration File
- How to Configure GRUB With YaST

What a Boot Manager Is

To boot a system, you need a program that can boot the respective operating system. This program, called the *boot loader*, loads the operating system kernel, which then loads the system.

After running the Power-On Self Test (POST), the PC BIOS searches the boot sector on the first hard drive for a boot loader. If it finds one, it turns control of the boot process over to the boot loader.

The boot loader then locates the operating system files on the hard drive and starts the operating system.

A *boot manager* is also a boot loader, but it can handle several operating systems. If there is more than 1 operating system present, the boot manager presents a menu allowing you to select a specific operating system to be loaded.

After selecting an operating system, the boot manager loads the operating system files and specifies the kernel parameters.

Linux boot managers can be used to load Linux or other operating systems, such as Microsoft Windows. Windows NT, Windows 2000, or Windows XP.

A boot manager is designed with the following 2-stage architecture:

- **Stage 1.** The first stage of a boot manager is usually installed in the master boot record (MBR) of the hard disk (first stage boot loader).

  As the space in the MBR is limited to 446 bytes, this program code merely contains the information for loading the next stage.

  Stage 1 can be installed in the MBR, in the boot sectors of partitions, or on a floppy disk.

- **Stage 2.** This stage usually contains the actual boot manager. The files of the boot manager are located in the directory /boot/.

Boot Managers in SUSE Linux

SUSE Linux Enterprise Server provides 2 boot managers for the Linux environment: GRUB (GRand Unified Bootloader) and LILO (LInux LOader).

To understand something about these boot managers, you need to know the following:

- GRUB Boot Manager
- LILO Boot Manager
- Map Files, GRUB, and LILO
- Additional Information

GRUB Boot Manager

GRUB is the standard boot manager in SUSE Linux Enterprise Server. The following are some special features of GRUB:

- **File system support.** Stage 2 includes file system drivers for ReiserFS, ext2, ext3, Minix, JFS, XFS, FAT, and FFS (BSD). For this reason, the boot manager can access files through filenames even before the operating system is loaded.

  For example, this feature is useful for searching for the kernel and loading it if the boot manager configuration is faulty.

- **Interactive control.** GRUB has its own shell that enables interactive control of the boot manager.

LILO Boot Manager

Because LILO is not the default boot manager of SUSE Linux Enterprise Server, it is only covered briefly in this objective.

The LILO configuration file is /etc/lilo.conf. Its structure is similar to that of the GRUB configuration file.

When you modify the configuration file /etc/lilo.conf, you need to enter the command **lilo** for the changes to be applied.

You also need to use the command lilo when moving the kernel or the initrd on your hard disk.

Map Files, GRUB, and LILO

The main obstacle for booting an operating system is that the kernel is usually a file within a file system on a partition on a disk. These concepts are unknown to the BIOS. To circumvent this, maps and map files were introduced.

These maps simply note the physical block numbers on the disk that comprise the logical files. When such a map is processed, the BIOS loads all the physical blocks in sequence as noted in the map, building the logical file in memory.

In contrast to LILO, which relies entirely on maps, GRUB tries to become independent from the fixed maps at an early stage. GRUB achieves this by means of the file system code, which enables access to files by using the path specification instead of the block numbers.

Additional Information

You can refer to the following sources for additional information on GRUB and LILO:

- **Linux system.** The following are available from your Linux system:
  - Enter the following for manual pages and info files:
    - info grub
    - man grub
    - man grub-install
    - man grub-md5-crypt
    - man lilo
    - man 5 lilo.conf
  - Check the following README files:
    - In the directory /usr/share/doc/packages/grub/
    - In the directory /usr/share/doc/packages/lilo/
- **Internet sites.** Check the following site on the Internet:
  - http://www.gnu.org/software/grub/

How to Start the GRUB Shell

Because GRUB has its own shell, you can boot the system manually if the Linux system does not start due to an error in the boot manager.

There are 2 ways to start the GRUB shell:

- From a Running System
- From the Boot Prompt

From a Running System

To start the GRUB shell during operation, enter the command grub as root. The following appears:

```
    GNU GRUB  version 0.94  (640K lower / 3072K upper memory)

 [ Minimal BASH-like line editing is supported.  For the first word, TAB
   lists possible command completions.  Anywhere else TAB lists the
   possible completions of a device/filename. ]

grub>
```

As in a Bash shell, you can complete GRUB shell commands with the Tab key. To find out which partition contains the kernel, enter the command find, as in the following:

```
grub> find /boot/vmlinuz
 (hd0,2)

grub>
```

In this example, the kernel (/boot/vmlinuz) is located in the third partition of the first hard disk (hd0,2).

You can close the GRUB shell by entering **quit**.

From the Boot Prompt

You can start the GRUB shell at the boot prompt by doing the following:

1. From the graphical boot selection menu, press **Esc**.

   A text-based menu appears.

2. Start the GRUB shell by typing **c** (US keyboard layout).

How to Modify the GRUB Configuration File

You can configure GRUB by editing the file /boot/grub/menu.lst. The following is the general structure of the file:

- First, general options (such as the background color of the boot manager menu) are listed:

  ```
  color white/blue black/light-gray
  ```

- The general options are followed by options for the various operating systems that can be booted with the GRUB.

  Each entry for an operating system begins with the command title:

  ```
  title linux
       kernel (hd0,0)/boot/vmlinuz root=/dev/hda1
       initrd (hd0,0)/boot/initrd
  ```

The following is a simple example of the configuration file /boot/grub/menu.lst:

```
default 0
timeout 8

title Linux
    kernel (hd0,0)/boot/vmlinuz root=/dev/hda1
    initrd (hd0,0)/boot/initrd
```

The following describe the settings:

- **default 0.** The first entry (numbering from 0) is the default boot entry that is started automatically if no other entry is selected with the keyboard.
- **timeout 8.** The default boot entry is started automatically after 8 seconds.

How to Configure GRUB With YaST

While you can use YaST (Boot Loader Configuration module) to simplify the configuration of the boot loader, you should not experiment with this module unless you understand the concepts behind it.

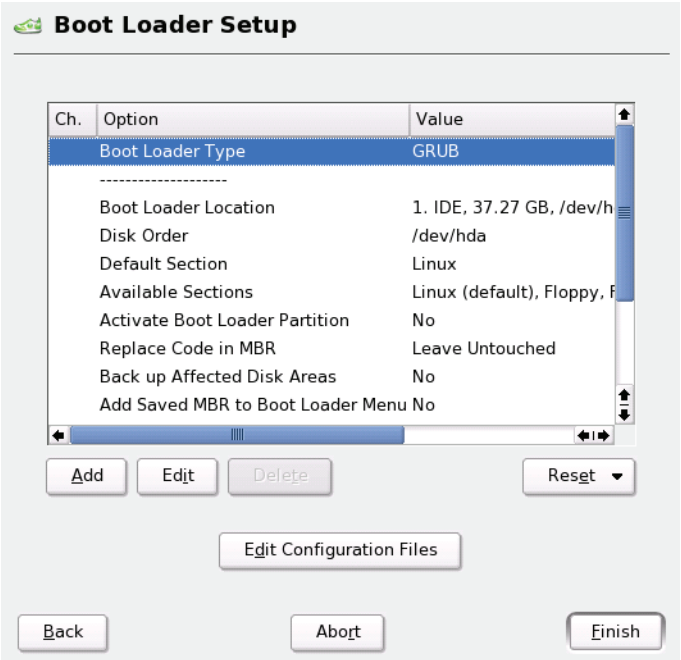
To configure GRUB with YaST, do the following:

1. From the KDE desktop, start the YaST Boot Loader Configuration module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > Boot Loader Configuration**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 bootloader**.

The following appears:

Figure 5-8



These are the current GRUB settings for your system, and includes a **Ch** (changed) column that indicates which default settings you have changed.

2. Do one of the following:
   - Add an option to the list by selecting **Add**, selecting an ***option*** from the drop-down list, and then selecting **OK**.

     or
   - Edit an option by selecting the ***option***, selecting **Edit**, and then changing the parameters in the displayed dialog.

     or
   - Delete an option by selecting the ***option*** and selecting **Delete**.

The following describes some of the more commonly-used GRUB options:

- **Boot Loader Type.** You can use this option to switch between GRUB and LILO. A dialog lets you specify the way this change should be performed.

  If you start the boot loader configuration in the running system, you can load the configuration from the hard disk.

  If you decide to return to the original boot loader, you can load its configuration by means of the last option. However, this is possible only if you do not close the boot loader module.

- **Boot Loader Location.** You can use this option to define where to install the boot loader in the MBR, in the boot sector of the boot partition (if available), or on a floppy disk.

  Use Others to specify a different location.

- **Disk Order.** If your computer has more than 1 hard disk, specify the boot sequence of the disks as defined in the BIOS setup of the machine.

- **Default Section.** You can use this option to set which kernel or operating system to boot by default (if no other entry is selected in the boot menu).

  The default system is booted after the timeout.

- **Available Sections.** The existing entries of the boot menu are listed under this option in the main window. If you select this option and then select **Edit**, a dialog opens that is identical to the Default Entry dialog.

- **Make Boot Loader Partition Active.** You can use this option to activate the partition whose boot sector holds the boot loader, independently from the partition that the directory with the helper files of the boot loader holds (the /boot/ or root directory /).

- **Replace Code in MBR.** You can use this option to specify whether to overwrite the MBR, which may be necessary if you have changed the location of the boot loader.
- **Backing up Files and Parts of Hard Disks.** This option backs up the changed hard disk areas.
- **Add Saved MBR to Boot Loader Menu.** This option adds the saved MBR to the boot loader menu.

3. (Optional) Display and edit the configuration files (such as /boot/grub/menu.lst or /boot/grub.conf) by selecting **Edit Configuration Files**.
4. (Optional) You can select from the following options by selecting one of the following from the **Reset** drop-down list:
   - **Propose New Configuration.** This option generates a new configuration suggestion. Older Linux versions or other operating systems found on other partitions are included in the boot menu, enabling you to boot Linux or its old boot loader. The latter takes you to a second boot menu.
   - **Start from Scratch.** This option lets you create the entire configuration from scratch. No suggestions are generated.
   - **Reread Configuration from Disk.** If you already performed some changes and are not satisfied with the result, you can reload your current configuration with this option.
   - **Propose and Merge with Existing GRUB Menus.** If another operating system and an older Linux version are installed in other partitions, the menu is generated from an entry for the new SUSE Linux, an entry for the other system, and all entries of the old boot loader menu.

     This procedure might take some time and is only available with GRUB.
   - **Restore MBR from Hard Disk.** The MBR saved on the hard disk is restored.

5. When you finish configuring the boot loader, save the configuration changes by selecting **Finish**.

Remember that the sequence of the options or commands is very important in GRUB. If the specified sequence is not followed, the machine might not boot.

Exercise 5-4 Manage the Bootloader

In this exercise, you do the following:

- Part I: Pass Kernel Parameters to the Bootloader
- Part II: Configure Boot Managers

Part I: Pass Kernel Parameters to the Bootloader

Do the following:

1. Reboot the system to runlevel 1, set the VGA mode to normal, and disable the SUSE splash screen by doing the following:
   a. From a terminal window on the KDE desktop, su to root (**su -**) with a password of **novell**.
   b. Switch to runlevel 6 by entering **init 6**.

      The system shuts down and reboots to a GUI menu that includes the options Linux and Failsafe.
   c. Stop the timer for the GUI menu by pressing the **Spacebar**.
   d. Exit the GUI menu by pressing **Esc**; then select **OK**.

      A TUI (Text User Interface) menu for GRUB appears with the same options (Linux and Failsafe).
   e. Make sure **Linux** is selected; then edit the boot commands by typing **e**.

      A menu with 2 boot commands (kernel and initrd) appears.
   f. Make sure that the command **kernel** is selected; then edit the command options by typing **e**.

      The command is displayed in a command line editor.
   g. Move to the beginning of the command by pressing **Home**.

h. Edit the command line to match the following:

kernel (hd0,1)/boot/vmlinuz root=/dev/hda2 vga=normal selinux=0 splash=0 resume=/dev/hda1 showopts 1

i. Save the edited command options by pressing **Enter**.

You are returned to the TUI menu for GRUB.

j. Boot the system to runlevel 1 by typing **b**.

To boot from the graphical screen, you could enter the options **1 vga=normal** (with a space before the "1") in the Boot Options field.

2. Log in by entering the root password of **novell**.
3. Check the previous and current runlevels by entering **runlevel**.

List the runlevels:

Table 5-6

Previous	Current

Note that both "1" and "S" indicate a single user mode.

4. Change to runlevel 3 by entering **init 3**.
5. Log in as **root** with a password of **novell**.
6. Check the previous and current runlevels by entering **runlevel**.

List the runlevels:

Table 5-7

Previous	Current

7. Change the default runlevel to 3 for the system in the file /etc/inittab:
   a. From the command line, enter **vim /etc/inittab**.
   b. Press the **Insert** key.
   c. Scroll down to the line id:5:initdefault: and make the following change:

      id:**3**:initdefault:
   d. Press **Esc**; then save the change and exit vim by entering **:wq**.
8. Reboot the system by entering **shutdown -r now**.
9. When the login prompt appears, log in as **root** with a password of **novell**.
10. Check to make sure your system booted to runlevel 3 by entering **runlevel**.
11. Change the default runlevel back to runlevel 5 using YaST:
    a. From the command line, enter **yast2 runlevel**.

       The Runlevel Editor appears.
    b. Select the Expert Mode by pressing **Alt+E**.

       The Runlevel Editor switches to the Expert mode.
    c. Press **Alt+D**; then from the drop-down list select **5: Full multiuser with network and xdm**.
    d. Save the changes by pressing **Alt+F** (to select **Finish**).

       You are returned to the command line.
12. Reboot the system with a 1 minute warning and an alert message to users by entering the following:

    shutdown -r 1 “==========> Log off now <==========”

    The message appears on your screen and the system reboots to the GUI login screen.
13. Log in as **geeko** with a password of **N0v3ll**.

Part II: Configure Boot Managers

Although GRUB is newer and was the first to overcome the 1024th cylinder problem, LILO still has many features that make it an attractive option.

You decide to test LILO by doing the following:

1. Change the boot manager to LILO:
   a. From the KDE desktop, start YaST by selecting the **YaST** icon.
   b. Enter a password of **novell**; then select **OK**.

      The YaST Control Center appears.
   c. Select **System > Boot Loader Configuration**.

      The Boot Loader Setup dialog appears.
   d. From the list of boot loader parameters, make sure that **Boot Loader Type** is selected; then select **Edit**.

      A boot loader type dialog appears.
   e. From the Boot Loader Type drop-down list, select **LILO**; then select **OK**.

      A warning dialog appears about changing your boot loader.
   f. Make sure **Convert Current Configuration** is selected; then select **OK**.
   g. Save the changes by selecting **Finish**.
2. Customize the LILO configuration (/etc/lilo.conf) to enhance video settings and secure LILO with a boot password:
   a. Open the file /etc/lilo.conf in an editor by pressing **Alt+F2** and entering **kdesu kate /etc/lilo.conf**; then enter a password of **novell** and select **OK**.
   b. (Conditional) If the parameter **vga = 791** does not exist at the top of the file under “boot = /dev/hda,” add the parameter.

This is a VESA frame buffer code in decimal format indicating that the screen should be set to 1024x768 and a 16-bit color depth.

c. Under the “root = /dev/hda2” line in the image section with “label = Linux,” add the parameter **password = novell**.

This parameter password protects this image at boot.

d. Save the changes by selecting **File > Save**.

Keep the file lilo.conf open in the Kate window.

3. Open a terminal window; then su to root (**su -**) with a password of **novell**.

4. Update the LILO configuration by entering **lilo**.

Now that the LILO configuration is updated with the new parameters, you should remove the password parameter to prevent others from viewing the password in the file /etc/lilo.conf.

5. From the Kate window, edit the file /etc/lilo.conf by deleting **password = novell**.

6. Save the change and close Kate by selecting **File > Quit**; then select **Save**.

7. From the terminal window, reboot the system by entering **reboot**.

When the system boots to the LILO startup screen, you are asked for a password.

8. Enter a password of **novell**; then select **OK**.

The system begins booting.

The GUI login screen appears.

9. Log in as **geeko** with a password of **N0v3ll**.

10. Test your KDE desktop (X windows) by starting and closing a few applications.

    Now that you've tested LILO, you decide to return to using GRUB as your boot loader.

11. Re-install GRUB as the boot loader for your SLES 9 server by editing the file /etc/sysconfig/bootloader:

    a. Open the file /etc/sysconfig/bootloader in an editor by pressing **Alt+F2** and entering **kdesu kate /etc/sysconfig/bootloader**; then enter a password of **novell** and select **OK**.

    b. Locate the parameter LOADER_TYPE="lilo" and change it to the following:

       LOADER_TYPE=**"grub"**

    c. Save the change and close the Kate window by selecting **File > Quit**; then select **Save**.

    d. Open a terminal window and su to root (**su -**) with a password of **novell**.

    e. Re-install the GRUB boot loader by entering **grub-install /dev/hda**.

12. Reboot the server by entering **init 6**.

13. From the GUI login screen, log in as **geeko** with a password of **N0v3ll**.

14. (Optional) If you have time, add a GRUB password using YaST (**System > Boot Loader Configuration**); then reboot and test the configuration.

(End of Exercise)

Objective 5 Modify System Settings

To tune your SUSE Linux Enterprise Server system to meet your specific requirements, you need to know how to do the following:

- View and Change System Settings (/proc/sys/)
- Modify Kernel and Hardware Parameters With Powertweak
- Configure /etc/sysconfig/ Files With YaST

View and Change System Settings (/proc/sys/)

The files and directories under the directory /proc/ contain a wealth of information about various aspects of the running system. This includes the files under /proc/sys/, which you can view and modify during operation to change the system settings.

The files in /proc/ and /proc/sys/ are not kept on the hard disk but are created by the kernel in memory when the system starts up. Changes to these files are lost after a reboot.

To manage these settings you need to know how to do the following:

- View the Current Configuration
- Edit the Current Configuration

View the Current Configuration

The individual configuration files in the directory /proc/sys/ are text files that you can view with commands such as cat and less.

For example, entering **cat /proc/sys/dev/cdrom/info** displays the following information about the CD-ROM drive:

```
geeko@DA50:~> cat /proc/sys/dev/cdrom/info
CD-ROM information, Id: cdrom.c 3.20 2003/12/17

drive name:             hdc
drive speed:            0
drive # of slots:       1
Can close tray:         0
Can open tray:          0
Can lock tray:          1
Can change speed:       1
Can select disk:        0
Can read multisession:  1
Can read MCN:           1
Reports media changed:  1
Can play audio:         1
Can write CD-R:         0
Can write CD-RW:        0
Can read DVD:           1
Can write DVD-R:        0
Can write DVD-RAM:      0
Can read MRW:           0
Can write MRW:          0
Can write RAM:          0
```

You can also use the command sysctl to view all or specific modifiable values below /proc/sys/, as in the following:

```
geeko@DA50:~> /sbin/sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0

geeko@DA50:~> /sbin/sysctl -a
sunrpc.tcp_slot_table_entries = 16
sunrpc.udp_slot_table_entries = 16
sunrpc.nlm_debug = 0
sunrpc.nfsd_debug = 0
sunrpc.nfs_debug = 0
sunrpc.rpc_debug = 0
abi.fake_utsname = 0
abi.trace = 0
abi.defhandler_libcso = 68157441
...
```

Edit the Current Configuration

You can use the command echo to edit individual configuration values. For example, entering the following command activates routing:

echo 1 > /proc/sys/net/ipv4/ip_forward

You can also do the same thing by using the command sysctl:

sysctl -w net.ipv4.ip_forward=1

Another example is deploying an Oracle database. This requires a number of kernel parameters to be set, as in the following:

```
DA50:~ # echo 65535 > /proc/sys/fs/file-max
DA50:~ # echo 2147483648 > cat /proc/sys/kernel/shmmax
```

The same is true of using sysctl to deploy an Oracle database:

```
DA50:~ # sysctl -w fs.file-max=65535
DA50:~ # sysctl -w kernel.shmmax=2147483648
```

If you want to load a number of kernel parameters when the system is booted, you can use the command sysctl. You enter the parameters in the file /etc/sysconfig/sysctl. The following are some sample settings:

```
net.ipv4.ip_forward = 1
net.ipv4.icmp_echo_ignore_broadcasts = 1
fs.file-max = 65535
kernel.shmmax = 2147483648
```

Once you finish editing the configuration file, you set the parameters by entering **sysctl -p** or **/etc/init.d/boot.sysctl start**.

To execute the script /etc/init.d/boot.sysctl when the system is booted, activate it by entering **insserv -d boot.sysctl**.

Modify Kernel and Hardware Parameters With Powertweak

SuSE Linux Enterprise Server offers a special tool for configuring kernel and hardware parameters called *Powertweak*. This tool includes the daemon powertweakd and a graphical YaST front-end.

A significant advantage of using Powertweak to set kernel and hardware parameters is that a short description is provided for every parameter.

To start Powertweak, do the following:

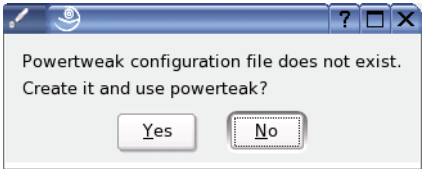
1. From the KDE desktop, start the YaST Powertweak Configuration module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > Powertweak Configuration**.

or

- Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 powertweak**.

The first time you start Powertweak, the following appears:

Figure 5-9



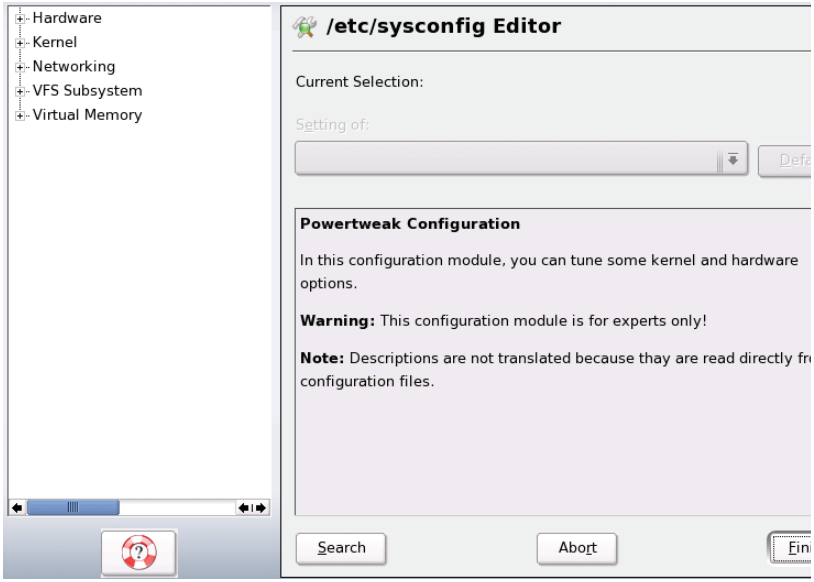
2. Create the Powertweak configuration file by selecting **Yes**.

   The configuration file /etc/powertweak/tweaks is created and the daemon is started. From this point on, the daemon is started every time the system is booted

   The links to the start script /etc/init.d/powertweakd are also set in the respective runlevel directories under /etc/init.d/.

The following appears:

Figure 5-10



3. To find a parameter, do one of the following:
   - From the left frame, expand a category and subcategories until you find the parameter you want to change; then select the ***setting***.

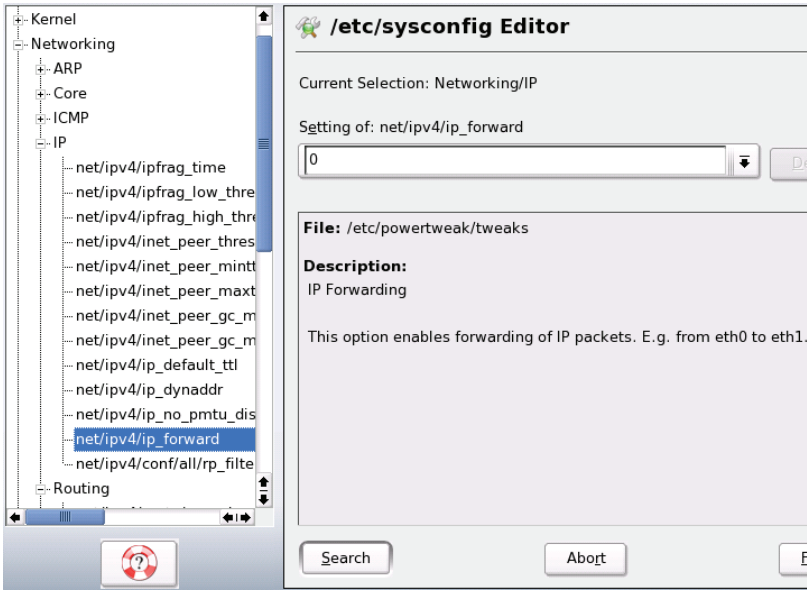
     or

   - Select **Search** and enter a ***keyword***; then select **OK**.

   Once you select a parameter, information appears in the right frame.

For example, if you find and select **Networking > IP > net/ipv4/ip_forward**, the following appears:

Figure 5-11



4. From the right frame, read the information (file, possible values, default value, and description); then select or enter the ***setting***.

   For example, you can activate routing by entering **1**.

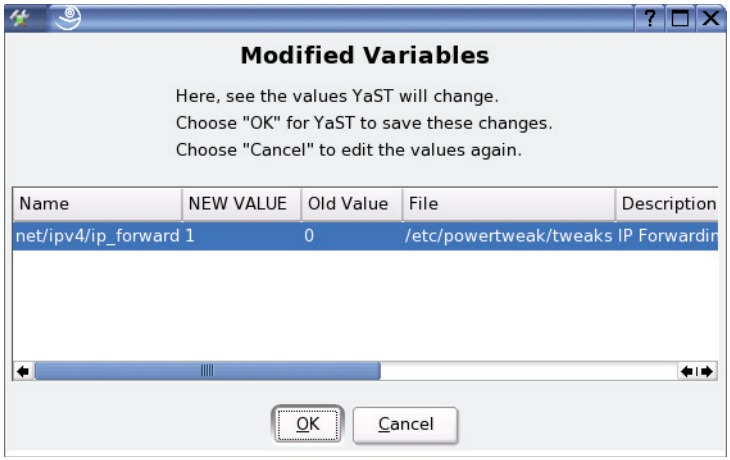
5. (Optional) Find and configure other settings.

   If you want to return a setting to its default value, select **Default**.

6. When you finish, select **Finish**.

The following appears:

Figure 5-12



This dialog lists a summary of all the changes you have indicated that you want made.

7. When you finish reviewing the list, save the changes by selecting **OK**.

   The changes are saved and activated by SuSEConfig.

Configure /etc/sysconfig/ Files With YaST

All changes to the system configuration you make with YaST happen in 1 of the following ways:

- **Direct modification of configuration files.** The following are some examples:
  - Installation of software resulting in changes to the RPM database
  - Printer configuration is written directly to the configuration files of the CUPS print system (in /etc/cups/)

- Using the Runlevel Editor modifies /etc/inittab and the links in the runlevel directories in /etc/init.d

- **YaST modifies the configuration files in /etc/sysconfig/.** The following are some examples:
  - The network configuration is saved in files in the directory /etc/sysconfig/network/.
  - The mail configuration is saved in /etc/sysconfig/mail and /etc/sysconfig/postfix.
  - The DMA mode for hard disks is saved in /etc/sysconfig/hardware.

A majority of configuration settings for the SUSE Linux Enterprise Server environment are saved in files in the directory /etc/sysconfig/.

You can edit these files by using YaST modules (configuration saved automatically when you select Finish), by using a text editor, or by using the YaST module /etc/sysconfig Editor.

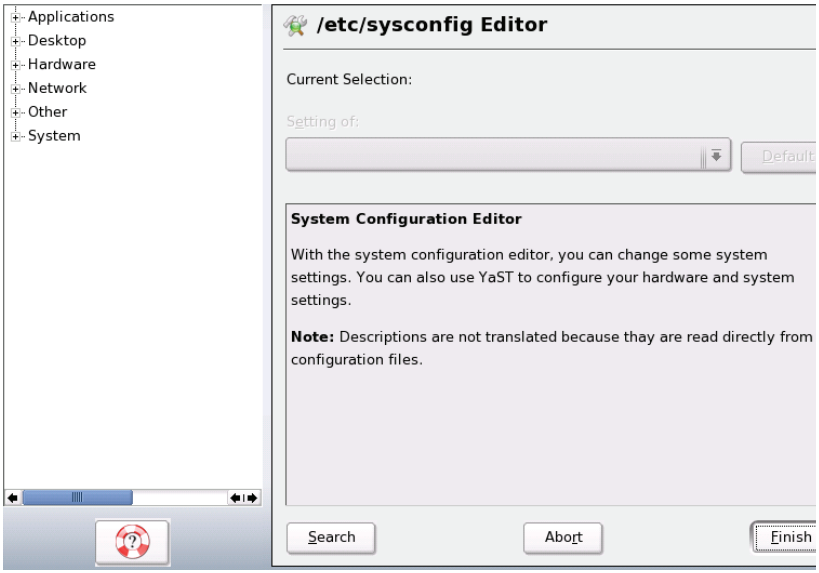
To configure the files in /etc/sysconfig/ and its subdirectories with YaST, do the following:

1. From the KDE desktop, start the YaST /etc/sysconfig Editor module by doing one of the following:
   - Select the **YaST** icon, enter the root ***password***, and select **OK**; then select **System > /etc/sysconfig Editor**.

     or
   - Open a terminal window and enter **sux -** and the root ***password***; then enter **yast2 sysconfig**.

The following appears:

Figure 5-13



From this dialog, you can change the system settings stored in the directory /etc/sysconfig/.

2. To find a setting, do one of the following:
   - From the left frame, expand a category and subcategories until you find the setting you want to change; then select the ***setting***.

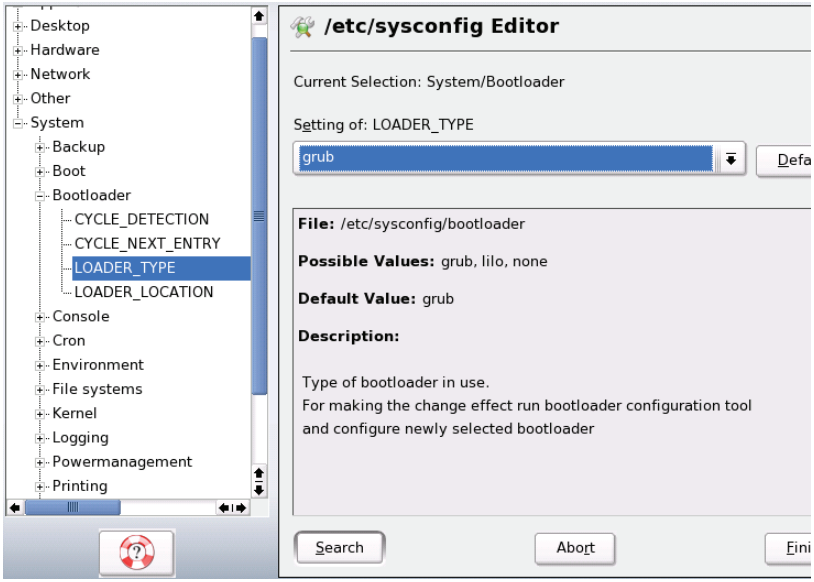
     or

   - Select **Search** and enter a ***keyword***; then select **OK**.

   Once you select a setting, information appears in the right frame.

For example, if you find and select **System > Bootloader > LOADER_TYPE**, the following appears:

Figure 5-14

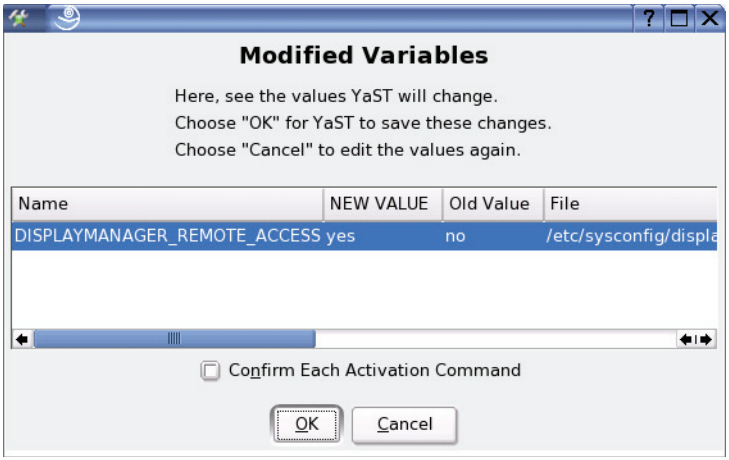


3. From the right frame, read the information (file, possible values, default value, and description); then select or enter the ***setting***.
4. (Optional) Find and configure other settings.

   If you want to return a setting to its default value, select **Default**.
5. When you finish, select **Finish**.

The following appears:

Figure 5-15



This dialog summarizes the changes you have indicated that you want made to the files in /etc/sysconfig/.

6. (Optional) If you want YaST to stop for confirmation before making each change, select **Confirm Each Activation Command**.

7. When you finish reviewing the list, save the changes by selecting **OK**.

   The changes are saved and activated by SuSEConfig.

Exercise 5-5 Change the Power Setting With YaST

Do the following:

1. From a terminal window, check the currently running CPU MHz speed by entering **cat /proc/cpuinfo**.

   The model name indicates the installed CPU and supported maximum processing speed (on most newer computers). The CPU MHz parameter indicates the current running speed.

   If there is a significant discrepancy between the maximum process speed and the CPU MHz value, then your CPU supports dynamic frequency scaling.

   This is controlled by the SLES 9 Power Management scheme and can be reconfigured to Maximum Performance.

   If you do not see a supported maximum processing speed with the model name, or do not notice a significant discrepancy, you can still complete the exercise to learn how to adjust the power settings.

2. From the KDE desktop, select the **YaST** icon; then enter a password of **novell** and select **OK**.

   The YaST Control Center appears.

3. Select **System > /etc/sysconfig Editor**.

   The /etc/sysconfig Editor appears.

4. From the list on the left, expand **System > Powermanagement > Scheme > Performance**; then select **POWERSAVE_CPUFREQUENCY**.

5. From the Setting of POWERSAVE_CPUFREQUENCY drop-down list (on the right), select **performance**.

6. Save the setting by selecting **Finish**; then select **OK**.

7. From the terminal window, su to root (**su -**) and enter a password of **novell**.

8. Restart the powersave daemon by entering **rcpowersaved restart**.

9. Check the currently running CPU MHz speed by entering **cat /proc/cpuinfo**.

   If they did not before, both the supported maximum processing speed and the CPU MHz value now closely match each other.

10. Close all windows.

(End of Exercise)

Summary

Objective	Summary
1. Describe the Linux Load Procedure	In this objective, you learned the following about the basic steps of booting a computer with a Linux system: ■ BIOS and Boot Manager ■ Kernel ■ initrd and linuxrc ■ init
2. Manage Runlevels	In this objective you continued learning about the initialization process and how to manage the runlevels associated with the services initialized during the process. To manage runlevels, you learned the following: ■ Runlevel Basics ■ How to Change the Runlevel at Boot ■ How to Manage Runlevels From the Command Line ■ How to Shut Down or Halt the System ■ How to Set Runlevels With YaST

Objective	Summary
3. Manage the Kernel	The primary function of the Linux kernel is to manage the system hardware resources, making them available to various system processes. To manage the kernel, you learned about the following: ■ Kernel Module Basics ■ How to Find Hardware Driver Information ■ How to Manage Modules From the Command Line ■ modprobe Configuration File (/etc/modprobe.conf) ■ Kernel Module Loader (kmod)
4. Manage the GRUB Boot Loader	To manage the GRUB boot loader, you learned about the following: ■ What a Boot Manager Is ■ Boot Managers in SUSE LINUX ■ How to Start the GRUB Shell ■ How to Modify the GRUB Configuration File ■ How to Configure GRUB With YaST

Objective	Summary
5. Modify System Settings	To tune your SUSE Linux Enterprise Server system to meet your specific requirements, you learned how to do the following: ■ View and Change System Settings (/proc/sys/) ■ Modify Kernel and Hardware Parameters With Powertweak ■ Configure /etc/sysconfig/ Files With YaST

Index

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z