

SUSE Linux Enterprise Desktop

11

www.novell.com

2009 1 15

管理指南



管理指南

所有内容的版权都属于 © 2006- 2009 Novell, Inc.

法律声明

本手册受 Novell 知识产权的保护。复制、复印或分发本手册，表示您明确同意遵守本许可协议的条款。

本手册可如上或作为捆绑包的一部分免费复制、复印或分发（电子和/或打印格式），前提是满足以下条件：

本版权声明及作者和贡献者姓名清晰明确地出现在复制、复印和分发的所有副本上。复制和/或分发本手册（尤其是打印格式）仅限于非商业用途。将本手册或其一部分用于任何其他用途，都必须事先获得 Novell, Inc 的明确授权。

有关 Novell 商标的列表，请参见 Novell 商标和服务标记列表

(<http://www.novell.com/company/legal/trademarks/tmlist.html>) [<http://www.novell.com/company/legal/trademarks/tmlist.html>]* Linux 是 Linus Torvalds 的注册商标。所有第三方商标均属其各自所有者的财产。商标符号（®、™ 等）代表 Novell 商标；星号 (*) 代表第三方商标。

本指南力求涵盖所有细节。但这并不确保本指南准确无误。无论是 Novell, Inc.、SUSE LINUX 产品 GmbH、作者还是翻译人员都不对任何可能的错误或因错误造成的任何后果负责。

目录

关于本指南	ix
部分 I 支持任务和常见任务	1
1 YaST 联机更新	3
1.1 使用 Qt 界面手动安装增补程序	4
1.2 使用 gtk 界面手动安装增补程序	5
1.3 自动联机更新	7
2 收集用于支持的系统信息	9
2.1 Novell Support Link 概述	9
2.2 使用 supportconfig	9
2.3 向 Novell 提交信息	11
2.4 更多信息	13
3 文本方式的 YaST	15
3.1 在模块中导航	16
3.2 组合键的限制	17
3.3 YaST 命令行选项	18
4 使用命令行工具管理软件	21
4.1 使用 Zypper	21
4.2 RPM — 包管理器	26

5	通过 Nomad 访问远程桌面	37
5.1	Nomad 先决条件	38
5.2	安装和设置	39
5.3	使用 Nomad	40
5.4	查错	40
5.5	更多信息	41
6	Bash 和 Bash 脚本	43
6.1	什么是“外壳”?	43
6.2	编写外壳脚本	48
6.3	重定向命令事件	49
6.4	使用别名	50
6.5	在 Bash 中使用变量	51
6.6	将命令分组和组合	53
6.7	使用通用流程构造语句	54
6.8	更多信息	55
	部分 II 系统	57
7	64 位系统环境中的 32 位和 64 位应用程序	59
7.1	运行时支持	59
7.2	软件开发	60
7.3	Biarch 平台上的软件编译	60
7.4	内核规范	61
8	引导和配置 Linux 系统	63
8.1	Linux 引导进程	63
8.2	init 进程	66
8.3	通过 /etc/sysconfig 配置系统	74
9	引导加载程序 GRUB	77
9.1	通过 GRUB 引导	78
9.2	使用 YaST 配置引导加载程序	85
9.3	卸载 Linux 引导加载程序	91
9.4	创建引导 CD	91
9.5	图形 SUSE 屏幕	92
9.6	查错	93
9.7	有关详细信息	94

10	特别的系统功能组件	95
10.1	特殊软件包的相关信息	95
10.2	虚拟控制台	102
10.3	键盘映射	102
10.4	语言和国家/地区特定的设置	103
11	打印机操作	107
11.1	打印系统工作流程	108
11.2	连接打印机的方法和协议	109
11.3	安装软件	109
11.4	网络打印机	110
11.5	图形打印接口	112
11.6	从命令行打印	113
11.7	SUSE Linux Enterprise Desktop 中的特殊功能	113
11.8	查错	116
12	使用 udev 进行动态内核设备管理	123
12.1	/dev目录	123
12.2	内核 uevents 和 udev	124
12.3	驱动程序、内核模块和设备	124
12.4	引导和启动设备设置	125
12.5	监视正在运行的 udev 守护程序	125
12.6	使用 udev 规则影响内核设备事件处理	126
12.7	永久设备命名	133
12.8	udev 使用的文件	133
12.9	有关详细信息	134
13	X Window 系统	135
13.1	手动配置 X Window 系统	135
13.2	安装和配置字体	141
13.3	更多信息	146
14	使用 FUSE 访问文件系统	147
14.1	配置 FUSE	147
14.2	装入 NTFS 分区	147
14.3	使用 SSHFS 装入远程文件系统	148
14.4	装入 ISO 文件系统	149
14.5	可用 FUSE 插件	149
14.6	更多信息	149

部分 III 移动计算机	151
15 Linux 中的移动计算	153
15.1 便携式计算机	153
15.2 移动硬件	160
15.3 手提电话和 PDA	161
15.4 更多信息	161
16 电源管理	163
16.1 省电功能	163
16.2 ACPI	164
16.3 硬盘的休眠	168
16.4 查错	169
16.5 更多信息	171
17 使用 Tablet PC	173
17.1 安装 Tablet PC 包	174
17.2 配置手写板设备	175
17.3 使用虚拟键盘	176
17.4 旋转显示器	176
17.5 使用手势识别	177
17.6 用手写笔记录和绘制	179
17.7 查错	181
17.8 更多信息	182
部分 IV 服务	185
18 基本联网知识	187
18.1 IP 地址和路由	190
18.2 IPv6 下一代因特网	192
18.3 名称解析	199
18.4 使用 YaST 配置网络连接	201
18.5 NetworkManager	218
18.6 手动配置网络连接	219
18.7 作为拨号助手的 smpppd	232
19 无线通讯	235
19.1 无线 LAN	235

20	网络中的 SLP 服务	245
20.1	安装	245
20.2	激活 SLP	246
20.3	SUSE Linux Enterprise Desktop 中的 SLP 前端	246
20.4	通过 SLP 提供服务	246
20.5	有关详细信息	247
21	使用 NTP 同步时间	249
21.1	使用 YaST 配置 NTP 客户机	249
21.2	在网络中手动配置 ntp	252
21.3	设置本地参考时钟	252
22	使用 NetworkManager	255
22.1	NetworkManager 的用例	255
22.2	启用 NetworkManager	256
22.3	配置网络连接	256
22.4	使用 KDE NetworkManager 控件	258
22.5	使用 GNOME NetworkManager 小程序	258
22.6	NetworkManager 和 VPN	260
22.7	NetworkManager 和安全性	261
22.8	常见问题	262
22.9	查错	264
22.10	更多信息	265
23	Samba	267
23.1	术语	267
23.2	配置 Samba 服务器	268
23.3	配置客户机	269
23.4	将 Samba 用作登录服务器	269
23.5	有关详细信息	270
24	通过 NFS 共享文件系统	271
24.1	安装所需软件	271
24.2	使用 YaST 导入文件系统	271
24.3	手动导入文件系统	272
24.4	采用 Kerberos 的 NFS	274
24.5	更多信息	274

25	文件同步	277
25.1	可用的数据同步软件	277
25.2	选择程序时的决定性因素	278
25.3	CVS 简介	281
25.4	rsync 简介	284
25.5	有关详细信息	285

关于本指南

本指南设计为由专业网络和系统管理员在操作 SUSE® Linux Enterprise 的过程中使用。同样，本指南旨在确保 SUSE Linux Enterprise 正确配置并且网络上的必需服务可用，使其在初始安装时正常运行。本指南不包含用于确保 SUSE Linux Enterprise 与用户企业的应用程序软件兼容或者其核心功能符合那些要求的过程。它假定已经进行了对完全要求的审计、已经请求安装或者已经请求用于此类审计的测试安装。

本指南包含如下内容：

管理

SUSE Linux Enterprise 提供了大量工具，用于自定义系统的各个方面。本部分介绍其中几个。

系统

通过研究本部分了解关于底层操作系统的更多信息。SUSE Linux Enterprise 支持许多硬件体系结构，您可以利用这点调试自己的应用程序，使之在 SUSE Linux Enterprise 上运行。引导加载程序和引导过程信息有助于您了解 Linux 系统的工作方式以及您自己的自定义脚本和应用程序与该系统的调和方式。

移动计算

便携式计算机、移动设备（如 PDA 或手机）和 SUSE Linux Enterprise 之间的通讯需要特别注意。要注意省电以及将不同设备集成到不断变化的网络环境中。同时要了解提供所需功能的后台技术。

服务

SUSE Linux Enterprise 被设计为一个网络操作系统。SUSE® Linux Enterprise Desktop 包含对许多网络服务的客户端支持。它可以很好地集成到包括 MS Windows 客户机和服务器在内的异构环境中。

本手册中的许多章节包含到附加文档资源的链接。这包括系统上提供的附加文档以及因特网上提供的文档。

有关该产品可用文档的概述和最新文档更新，请参见 <http://www.novell.com/documentation>。

1 可用文档

我们以不同的语言提供了这些手册的 HTML 和 PDF 版本。为用户和管理员提供了以下本产品的相关手册：

GNOME 用户指南 (↑*GNOME 用户指南*)

介绍 SUSE Linux Enterprise Desktop 的 GNOME 桌面。指导您使用和配置桌面并帮助您执行关键任务。它主要用于想要在日常生活中有效使用 GNOME 桌面作为默认桌面的最终用户。

应用程序指南 (↑*应用程序指南*)

了解如何使用和配置 SUSE Linux Enterprise Desktop 上的关键桌面应用程序。本指南介绍浏览器和电子邮件客户端以及办公应用程序和协作工具。还包括图形和多媒体应用程序。

部署指南 (↑*部署指南*)

显示如何安装单个或多个系统，以及如何利用产品继承功能建立部署基础结构。有各种方法可供选择，可以选择使用本地安装或网络安装服务器，也可以选择使用远程控制、高度自定义的自动安装技术进行大规模部署。

管理指南 [1]

讲述系统管理任务，如维护、监视和自定义初始安装的系统。

安全指南 (↑*安全指南*)

介绍系统安全的基本概念，包括本地安全方面和网络安全方面。显示如何利用诸如 Novell AppArmor 的产品继承安全软件（允许您按程序指定该程序可以读、写和执行哪些文件）或审计系统（可靠地收集关于任何安全相关事件的信息）。

系统分析和微调指南 (↑*系统分析和微调指南*)

关于问题检测、解决和优化的管理员指南。了解如何使用监视工具检查和优化系统以及如何有效管理资源。还包含常见问题和解决方法的概述以及其他帮助和文档资源。

使用 Xen 实现虚拟化 (↑*使用 Xen 实现虚拟化*)

提供了有关该产品虚拟化技术的简介。它是对应用程序各个字段以及 SUSE Linux Enterprise Server 支持的每个平台安装类型的概述，以及对安装过程的简短描述。

除了综合性手册，还提供几个快速入门指南：

安装快速入门(↑安装快速入门)

列出系统要求，并指导您从 DVD 或 ISO 映像逐步安装 SUSE Linux Enterprise Desktop。

Linux 审计快速入门

概述如何启用和配置审计系统以及如何执行关键任务（如设置审计规则、生成报告和分析日志文件）。

Novell AppArmor 快速入门

帮助您了解 Novell® AppArmor 背后的主要概念。

在 `/usr/share/doc/manual` 下的已安装系统中或者桌面的帮助中心中查找大多数 SUSE Linux Enterprise Desktop 手册的 HTML 版本。在 <http://www.novell.com/documentation>（您可从该处下载产品手册的 PDF 或 HTML 版本）上查找最新的文档更新。

2 反馈

提供了多种反馈渠道：

- 要报告产品组件的错误或要提交增强请求，请使用 <https://bugzilla.novell.com/>。如果您刚开始使用 Bugzilla，您会发现 *bug* 书写常见问题可能很有用，该功能可以从 Novell Bugzilla 主页中找到。
- 我们希望听到您对本手册和本产品中包含的其他文档的意见和建议。请使用每页联机文档底部的用户意见功能并发表您的意见。

3 文档约定

以下是本手册中使用的版式约定：

- `/etc/passwd`：文件名和目录名
- `placeholder`：将 `placeholder` 替换为实际值
- `PATH`：环境变量 `PATH`

- `ls`、`--help`：命令、选项和参数
- 用户：用户和组
- `Alt`、`Alt + F1`：按键或组合键；这些键以大写形式显示，如在键盘上一样
- 文件，文件 > 另存为：菜单项，按钮
- 跳舞的企鹅（企鹅一章，↑其他手册）：这是对其他手册中的章节的引用。

部分 I. 支持任务和常见任务

YaST 联机更新

Novell 会一直为您的产品提供软件安全性更新。默认情况下，openSUSE Updater 用于使您的系统保持最新状态。请参见第 7.5 节 “Keeping the System Up-to-date” (第 7 章 *Installing or Removing Software*, ↑部署指南) 以获取有关 openSUSE Updater 的进一步信息。本章涵盖用于更新软件包的备用工具：YaST 联机更新。

更新软件安装源提供了 SUSE® Linux Enterprise Desktop 的当前增补程序。如果安装时已注册您的产品，则更新安装源已配置。如果未注册 SUSE Linux Enterprise Desktop，可通过运行 YaST 中的 **软件 > 联机更新配置** 来执行此操作，也可启动 **高级 > 注册** 以获取支持并获取更新安装源。或者，可以从信任的源中手动添加更新安装源。要添加或删除安装源，请使用 YaST 中的 **软件 > 软件安装源** 来启动安装源管理器。请在第 7.4 节 “Adding Software Repositories” (第 7 章 *Installing or Removing Software*, ↑部署指南) 中了解更多有关安装源管理器的内容。

注意: 访问更新编目时出错

如果您不能访问更新编目，可能是由于订购已过期。通常，SUSE Linux Enterprise Desktop 的订阅期为一年或三年，在此期间您可以访问更新编目。订购结束后，将拒绝您访问更新编目。

拒绝访问更新编目时，您将看到一条警告消息，建议您访问 Novell Customer Center 并检查您的订阅。可通过 <http://www.novell.com/center/> 访问 Novell Customer Center。

提供了不同相关级别的更新。安全性更新会修复严重的安全性危险，请务必安装。推荐更新会修复可能危及计算机安全的问题，而可选更新会修复与安全性无关的问题或提供增强功能。

要用 YaST 安装更新和改进，请从 YaST 运行软件> 联机更新。您系统当前可用的所有新增补程序（除了可选增补程序外）都已标记为安装。单击接受或应用会自动安装这些增补程序。安装完成后，用完成确认。您的系统现在已是最新了。

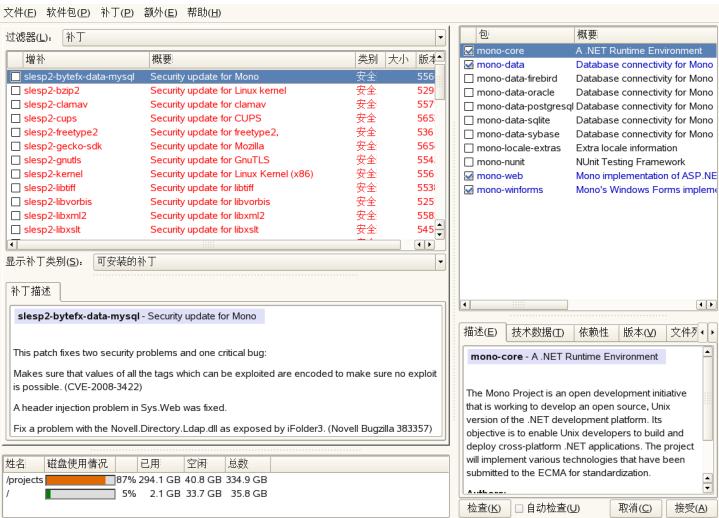
提示: 禁用 deltarpm

默认情况下，更新会作为 deltarpm 下载。由于从 deltarpm 重建 rpm 包是一项需要大量内存和 CPU 时间的任务，某些设置或硬件配置可能需要禁用 deltarpm 以提高性能。要禁用 deltarpm，请编辑文件 /etc/zypp/zypp.conf，并将 download.use_deltarpm 设置为 false。

1.1 使用 Qt 界面手动安装增补程序

联机更新窗口由四部分组成。左边是所有可用增补程序的列表。在增补程序列表下可找到选定增补程序的描述。右列列出了选定增补程序中所含的包（一个增补程序可由多个包组成），下面是选定包的详细描述。磁盘用量可以选择性地显示在左列底部（默认情况下，此显示处于淡出状态，使用点阵幻灯片可使其可见）。

图 1.1 YaST 联机更新



增补程序显示列出了 SUSE Linux Enterprise Desktop 的可用增补程序。增补程序是按安全相关性排序的。安全性、推荐和可选。有三个不同的增补程序视图。可以使用显示增补程序类别切换视图。

需要的增补程序（默认视图）

当前未安装的适用于系统上已安装的包的增补程序。

不需要的增补程序

适用于您的系统未安装包的增补程序或要求已满足的增补程序（因为已从另一源进行了更新）。

所有增补程序

SUSE Linux Enterprise Desktop 可用的所有增补程序。

列表项由符号和增补程序名称组成。要查看可能出现的符号的列表，请按 **Shift + F1**。安全性和建议增补程序需要的操作是自动预设置的。这些操作有 *Autoinstall*、*Autoupdate* 或 *Autodelete*。可选增补程序的操作没有预置 — 右键单击某个增补程序然后从列表中选择操作。

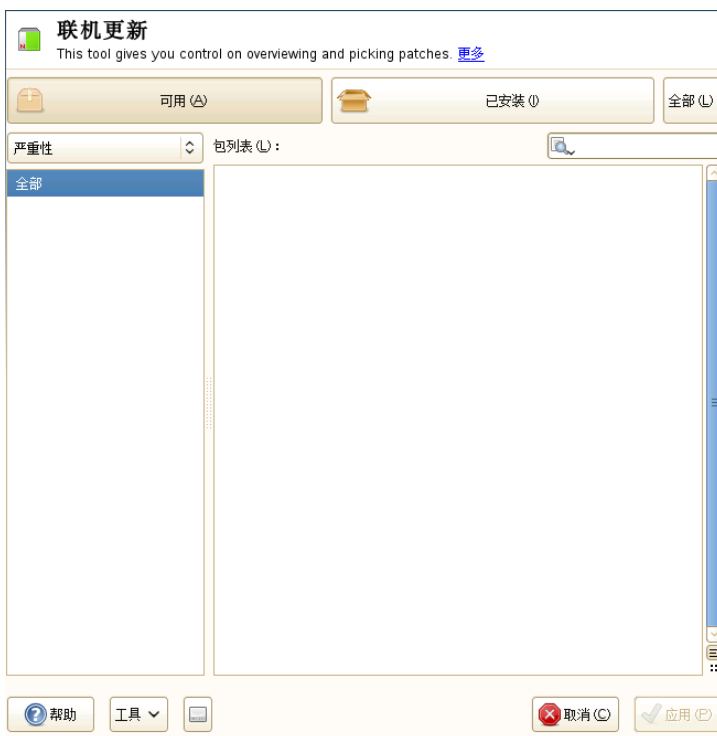
如果从非更新安装源的某个安装源安装最新包，此安装可能满足此包的某个增补程序的要求。在这种情况下，在增补程序摘要前会显示一个复选标记。该增补程序将显示在列表中，直到将其标记用于安装。这实际上不会安装增补程序（因为该包已经是最新的），而是将该增补程序标记为已安装。

多数增补程序包含几个包的更新。要更改单个包的操作，请在包窗口中右键单击某个包，并选择操作。按需要标记所有增补程序和包后，单击接受。

1.2 使用 gtk 界面手动安装增补程序

联机更新窗口由 2 个主要部分组成。左侧面板列出了所有增补程序并提供用于该增补程序列表的不同过滤器。查看右侧面板，获取应用它们后所实施更改的列表。

图 1.2 YaST 联机更新



增补程序列表过滤器

可用

当前未安装的适用于系统上已安装的包的增补程序。

“已安装”

已安装的增补程序。

所有

已安装或可用的增补程序。

严重性

仅显示可选、推荐或安全性增补程序。默认情况下，将显示所有增补程序。

安装源

该过滤器使您可以按安装源显示增补程序。

列出的包

在此应用您的自定义过滤器。

单击某个增补程序项将在左侧面板的底部区域打开有关此增补程序详细信息的一行。您可以在此处查看详细的增补程序描述以及可用的版本。您还可以选择安装可选增补程序（安全性和推荐增补程序已在安装时预选）。

1.3 自动联机更新

YaST 还能安装自动更新。打开软件 > 联机更新配置。选中 *自动联机更新* 并选择是要每天、每周还是每月更新一次。有些增补程序（如内核更新）需要用户交互，交互可能会导致自动更新停止。因此，如果希望更新过程完全自动执行，应选中 *跳过交互式增补程序*。执行此操作后，应不时地运行手动 *联机更新* 以便安装需要交互的增补程序。

收集用于支持的系统信息

发生问题时，可以使用 `supportconfig` 收集系统信息，如使用的当前内核版本、硬件、RPM 数据库、分区等。结果用于帮助 Novell 支持中心找出问题。

2.1 Novell Support Link 概述

Novell Support Link (NSL) 对 SUSE Linux Enterprise Desktop 来说是全新的。它是一种收集系统信息并允许您将此信息上载到另一台服务器以供进一步分析的工具。Novell 支持中心使用 Novell Support Link 从有问题的服务器收集系统信息，并将该信息发送到 Novell 的公用 FTP 服务器。收集的系統信息包括：当前正在使用的内核版本、硬件、RPM 数据库、分区等。该结果用于帮助 Novell 支持中心解决您的未决服务请求。

Novell Support Link 有两种使用方法：

1. 使用 YaST 支持模块，
2. 使用命令行实用程序 `supportconfig`。

YaST 支持模块调用 `supportconfig` 来收集系统信息。

2.2 使用 `supportconfig`

以下各节描述了如何通过 YaST 从命令行中使用 `supportconfig`，以及可以使用的选项。

2.2.1 使用 YaST 收集信息

要使用 YaST 收集系统信息，请如下操作：

- 1 打开 URL <http://www.novell.com/center/eservice> 并创建服务请求编号。
- 2 启动 YaST。
- 3 打开支持模块。
- 4 单击 *创建报告 tarball*。
- 5 从单选按钮列表选择一个选项。如果要先进行测试，请使用 *仅收集最少量的信息*。按下一步继续。
- 6 输入您的联系信息。使用 **步骤 1** [10] 中的服务请求编号，并将其输入标有 *Novell 11 位服务请求编号* 的文本字段中。按下一步继续。
- 7 即开始收集信息。该过程完成后，按下一步继续。
- 8 检查收集的数据，如果不需要某个文件名，请对相应的文件名使用 *从数据中删除*。按下一步继续。
- 9 保存 tarball。如果要上载到 Novell customer center，请确保将 *日志文件 tarball* 上载到 URL 中已激活。按下一步完成。

2.2.2 直接使用 supportconfig 收集信息

要从命令行使用 supportconfig，请如下操作：

- 1 打开外壳并转换为根用户。
- 2 无任何选项的情况下运行 supportconfig 会收集默认的系统信息。
- 3 等待工具完成。
- 4 默认的存档位置为 `/var/log`，文件名格式为 `nts_HOST_DATE_TIME.tbz`

2.2.3 常用的 supportconfig 选项

supportconfig 实用程序有多种启动选项。您可以通过 `supportconfig -h` 或使用手册页查看这些选项。一般，supportconfig 在无任何选项的情况下运行。以下是某些较常用启动选项的摘要。

- 使用最小选项 (`-m`) 来减少所收集的信息量。

```
supportconfig -m
```

- 在输出中包含附加的联系人信息（在一行中）：

```
supportconfig -E tux@example.org -N "Tux Penguin" -O "Penguin Inc." ...
```

- 针对某个问题进行查错时，可能您只想收集关于当前所研究问题领域的信息。例如，如果 LVM 出了问题，并且最近在默认 supportconfig 输出中发现了该问题。作出更改后，您想要收集最新的 LVM 信息。以下命令将收集最少的 supportconfig 信息，且仅 LVM 信息。

```
supportconfig -i LVM
```

要查看完整的功能列表，请运行：

```
supportconfig -F
```

- 通过相关的服务请求编号，使用 `-u` 和 `-r` 选项上载 supportconfig tar ball。例如，假设您已向 Novell 打开了一个服务请求，且跟踪编号为 12345678901，则运行以下命令：

```
supportconfig -ur 12345678901
```

2.3 向 Novell 提交信息

您可以使用 YaST 支持模块或 supportconfig 命令行实用程序向 Novell 提交系统信息。当发生服务器问题，且希望获得 Novell 帮助，您需要打开一个服务请求，然后向 Novell 提交服务器信息。以下描述了 YaST 和命令行两种方法。

过程 2.1 通过 YaST 向 Novell 提交信息

- 1 打开 URL <http://www.novell.com/center/eservice> 并创建服务请求编号。
- 2 写下 11 位服务请求编号。以下示例中将假设该服务请求编号为 12345678901。
- 3 在 YaST 支持模块窗口中单击创建报告 *tarball*。
- 4 选择使用自定义单选按钮。按下一步继续。
- 5 输入联系人信息，填写 Novell 11 位服务请求编号，并包含 Novell 的上载目标 URL。
 - 对于安全上载目标，请使用：<https://secure-www.novell.com/upload?appname=supportconfig&file={tarball}>。
 - 对于普通 FTP 上载目标，请使用：<ftp://ftp.novell.com/incoming>。

按下一步继续。信息收集开始。该过程完成后，按下一步继续。

- 6 查看数据集合，使用从数据中删除将删除要上载到 Novell 的 *tarball* 中排除的任何文件。按下一步继续。
- 7 默认情况下，*tarball* 的一个副本将保存到 `/root` 下。确认您使用的 Novell 上载目标为以上所述之一，且将日志文件 *tarball* 上载到 URL 中已激活。按下一步完成。
- 8 单击完成。

过程 2.2 通过 supportconfig 向 Novell 提交信息

- 1 打开 URL <http://www.novell.com/center/eservice> 并创建服务请求编号。
- 2 写下 11 位服务请求编号。以下示例中将假设该服务请求编号为 12345678901。
- 3 具有因特网连接的服务器：

3a 要使用默认上载目标，请运行：

```
supportconfig -ur 12345678901
```

3b 对于安全上载目标，请在同一行中使用以下命令：

```
supportconfig -r 12345678901 -U  
'https://secure-www.novell.com/upload?appname=supportconfig&file={tarball}'
```

4 不具有因特网连接的服务器：

4a 运行以下命令：

```
supportconfig -r 12345678901
```

4b 将 `/var/log/nts_SR12345678901*tbz tarball` 手动上载到 Novell 的 FTP 服务器 (<ftp://ftp.novell.com/incoming>)。

4c 还可以使用以下服务请求 URL 将该 tarball 挂接到您的服务请求：
<http://www.novell.com/center/eservice>。

5 此 tarball 在目录 <ftp://ftp.novell.com/incoming> 下之后，它将自动挂接到您的服务请求。

2.4 更多信息

请在以下文档中查找关于收集系统信息的更多信息：

- `man supportconfig` — `supportconfig` 的手册页
- `man supportconfig.conf` — `supportconfig` 配置文件的手册页
- <http://www.novell.com/communities/print/node/4097> — 使用 `supportconfig` 检查基本服务器运行状况
- <http://www.novell.com/communities/print/node/4827> — 创建自己的 `supportconfig` 插件

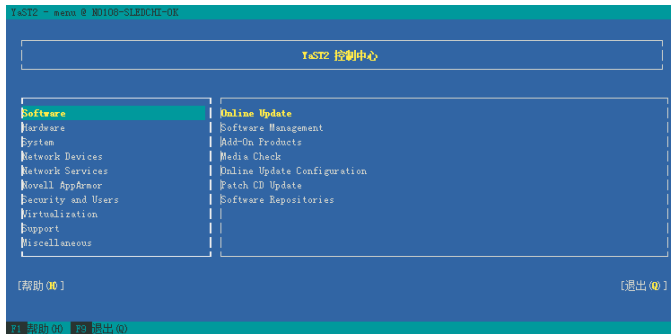
- <http://www.novell.com/communities/print/node/4800> — 创建中心 supportconfig 安装源

文本方式的 YaST

本节所针对的读者是在其系统上不运行 X 服务器而依赖于基于文本的安装工具的系统管理员和专家。它提供了与以文本方式启动和操作 YaST 有关的基本信息。

文本模式的 YaST 需要 ncurses 库以提供简单的伪图形用户界面。默认情况下已安装 ncurses 库。用于运行 YaST 的终端仿真器支持的最小大小为 80x25 个字符。

图 3.1 文本方式下 YaST 的主窗口



当以文本方式启动 YaST 时，将首先出现 YaST 控制中心，请参见图 3.1。该窗口包含三个区域。左框架有一个深白色边框，其中列出各个模块所属的类别。它使用有色背景来指示活动类别。具有狭窄的白色边框的右框架提供了活动类别中可用模块的概述。底部框架中包含帮助和退出按钮。

启动 YaST 控制中心时，将自动选择软件类别。使用↓键和↑键可更改类别。要从所选类别启动某个模块，请按→键。模块选择此时显示有深色边框。使用↓键和↑键可选择所需模块。按住箭头键在可用模块列表中滚动。模块选中时，模块标题将带有彩色背景。

按 Enter 键启动所需模块。模块中的各种按钮和选择字段中包含一个具有不同颜色（默认为黄色）的字母。使用 Alt + yellow_letter 可直接选择按钮，而无需使用 Tab 键导航到那里。通过按 Alt + Q 组合键或选择退出并按 Enter 退出 YaST 控制中心。

3.1 在模块中导航

下面在介绍 YaST 模块中的控制元素时，均假定所有功能键和 Alt 组合键都可用并且没有被指派不同的全局功能。有关可能出现的异常的信息，请参见第 3.2 节“组合键的限制”[17]。

在按钮和选择列表中导航

使用 Tab 键在按钮和包含选择列表的框架之间导航。要以相反顺序导航，请使用 Alt + Tab 键或 Shift + Tab 组合。

在选择列表中导航

使用箭头键（↑和↓）可浏览包含选择列表的活动框架中的各个元素。如果框架内的项超出了框架宽度，请使用 Shift + →或 Shift + ←来左右水平滚动。也可以使用 Ctrl + E 或 Ctrl + A。如果使用→或←键会导致更改活动框架或当前选择列表（像在控制中心中那样），则可以使用此组合键。

按钮、单选项按钮和复选框

要选择带空方括号（复选框）或空圆括号（单选按钮）的按钮，请按 Space 或 Enter 键。也可以直接使用 Alt + yellow_letter 来选择单选按钮和复选框。在这种情况下，无需使用 Enter 键进行确认。如果使用 Tab 键导航到某个项目，请按 Enter 键执行所选操作或激活相应的菜单项。

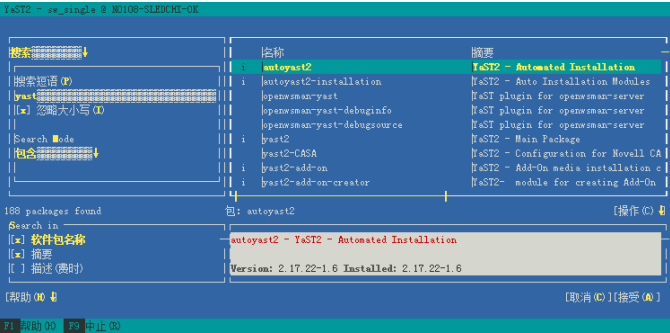
功能密钥

使用各功能键（F1 到 F12）可快速访问多个按钮。YaST 屏幕底部行中显示可用的 F 键快捷方式。功能键和按钮的实际映射关系取决于活动 YaST 模块，因为不同的模块提供不同的按钮（详细信息、信息、添加、删除等）。可以将 F10 用作接受、确定、下一步和完成。按 F1 可访问 YaST 帮助。

在 ncurses 方式中使用导航树

某些 YaST 模块使用窗口左侧的导航树选择配置对话框。在 ncurses 方式中，在导航树中进行选择之后必须按 **Enter**，才能显示所选对话框。这是一种有意行为，目的是在浏览导航树时避免耗时的重绘。

图 3.2 软件安装模块



3.2 组合键的限制

如果您的窗口管理器使用全局 **Alt** 组合键，则 YaST 中的 **Alt** 组合键可能无效。像 **Alt** 或 **Shift** 这样的键也可能被终端设置占用。

使用 **Esc** 代替 **Alt**

可以代替 **Alt** 而使用 **EscAlt** 快捷键。例如，**Esc - H** 可代替 **Alt + H**。（首先按 **Esc**，然后按 **H** 键。）

使用 **Ctrl + F** 和 **Ctrl + B** 执行向后和向前导航

如果 **Alt** 和 **Shift** 组合键由窗口管理器或终端占用，可改用组合键 **Ctrl + F**（向前）和 **Ctrl + B**（向后）。

功能键的限制

功能键也可用于执行多种功能。某些功能键可能会被终端占用而不能用于 YaST。但 **Alt** 组合键和功能键应该始终在纯文本控制台上完全可用。

3.3 YaST 命令行选项

除了文本方式界面之外，YaST 还提供了一个纯命令行界面。要获取 YaST 命令行选项列表，请输入：

```
yast -h
```

3.3.1 启动单个模块

为了节省时间，可以直接启动单个 YaST 模块。要启动模块，请输入：

```
yast <module_name>
```

要查看系统上所有可用模块名称的列表，请使用 `yast -l` 或 `yast --list`。例如，要启动网络模块，请输入 `yast lan`。

3.3.2 从命令行安装包

如果知道包名称且包是由您的任何活动安装安装源提供的，则可以使用命令行选项 `-i` 安装该包：

```
yast -i <package_name>
```

或

```
yast --install <package_name>
```

`package_name` 可以是一个简短的包名称（如 `gvim`，这是使用依赖性检查安装的），也可以是 `rpm` 包的全名（这是不进行依赖性检查安装的）。

如果需要具有 YaST 未提供的功能的，基于命令行的软件管理实用程序，请考虑使用 `zypper`。这个新实用程序使用相同的软件管理库，这也是 YaST 包管理器的基础。[第 4.1 节“使用 Zypper”](#) [21] 中涵盖了 `zypper` 的基本用法。

3.3.3 YaST 模块的命令行参数

为了在脚本中使用 YaST 功能，YaST 提供了对单个模块的命令行支持。并非所有模块都具有命令行支持。要显示某个模块的可用选项，请输入：

```
yast <module_name> help
```

如果模块不提供命令行支持，将以文本方式启动，并显示以下消息：

```
This YaST module does not support the command line interface.
```


使用命令行工具管理软件

本章描述 Zypper 和 RPM，这是两个用于管理软件的命令行工具。

4.1 使用 Zypper

Zypper 是用于安装和更新包的命令行工具。zypper 的语法类似于 `rug` 的语法。与 `rug` 相反，zypper 不需要在场景后运行 `zmd` 守护程序。有关 `rug` 兼容性的更多信息，请参见 http://en.opensuse.org/Zypper/Usage#Compatibility_with_Rug。这一点对于完成远程软件管理任务或从外壳脚本管理软件尤其有用。

zypper 在以下位置中包含有帮助概述：

```
zypper help
```

4.1.1 一般使用

zypper 的常用语法为：

```
zypper [global-options] command [command-options] [arguments] ...
```

不需要括在括号中的组件。执行 zypper 最简单的方式是，键入其名称后跟一个命令。例如，将所有需要的增补程序应用于系统类型：

```
zypper update
```

或者，可以从一个或多个全局选项中选择，方法是：只需在命令前面键入这些选项。例如，`--non-interactive` 表示运行此命令且不作任何询问，自行决定：

```
zypper --non-interactive update
```

要使用特定于某一特定命令的选项，请在此命令后面键入这些选项。例如，`--auto-agree-with-licenses` 表示将所有所需增补程序应用于系统且不要求确认任何许可证（所有这些许可证事先都已读取）：

```
zypper update --auto-agree-with-licenses
```

某些命令需要一个或多个自变量：

```
zypper install mplayer
```

某些选项也需要自变量。以下内容表示使用较新的包更新系统：

```
zypper update -t package
```

以上所有组合内容表示仅使用 `factory` 安装源安装，而且很冗长：

```
zypper -v install --repo factory mplayer amarok
```

也可使用 `+/-` 或 `~/!` 修改键一次性安装和删除包：

```
zypper install emacs -vim
```

Or:

```
zypper remove emacs +vim
```

或者，如果选择将 `-` 和指定的第一个包一起使用，则必须在该包之前写上 `--`，以免将其解释为命令选项：

```
zypper install -- -vim emacs
```

4.1.2 使用 Zypper 安装和删除软件

要安装来自注册安装源的包，请使用：

```
zypper install package_name
```

`zypper` 也支持通配符。要安装以 `package_name` 开头的所有包，请使用

```
zypper install package_name*
```

您还可以直接安装本地或远程 RPM — Zypper 还将根据以下命令自动安装所有 *package_name* 包：

```
zypper install http://www.example.com/package_name.rpm
```

为避免解决依赖性，请使用 `--no-recommends` 或 `--no-force-resolution`。

要删除已安装的包，请使用

```
zypper remove package_name
```

警告：请勿删除系统必需的包

请勿删除如 `glibc`、`zypper`、`kernel` 之类的包。这些包是系统必需的，如果缺少这些包，系统可能会停止工作。

默认情况下，在安装或删除选定包前 `zypper` 会要求进行确认。请使用 `--non-interactive` 选项覆盖此行为。在实际模式（安装、删除和更新）之前必须提供此选项，如同

```
zypper --non-interactive install package_name
```

此选项允许在脚本和 Cron 作业中使用 `zypper`。

如果要安装某包的对应源代码包，请使用

```
zypper source-install package_name
```

使用此命令，还可安装指定包的构建依赖性。如果不想执行此操作，请如下所示添加开关 `--no-build-deps`：

```
zypper source-install --no-build-deps package_name
```

当然，仅当您已将带有源代码包的安装源添加到安装源列表中时，此操作才有效。有关添加安装源的更多信息，请参见[第 4.1.4 节“管理安装源”](#) [24]。

更改已安装的软件库后，请校验是否仍满足所有的依赖性：

```
zypper verify
```

4.1.3 使用 Zypper 更新软件

使用 `zypper` 更新软件共有两种不同的方式。要将所有已正式发布的增补程序集成到您的系统中，只需运行

```
zypper update
```

这种情况下，会对安装源中的所有可用增补程序进行相关性检查，如有需要，还会安装这些增补程序。

如果某个安装源只有新包，但未提供增补程序，那么 `zypper update` 不会产生任何作用。要更新所有这些包，就必须进行指定，以安装 `package` 类型的更新：

```
zypper update -t package
```

要更新单个包，请使用安装命令：

```
zypper install package_name
```

可使用此命令来获取所有可用新包的列表：

```
zypper list-updates -t package
```

4.1.4 管理安装源

`zypper` 的所有安装或更新命令均基于 `zypper` 已知的安装源列表。要列出系统已知的所有安装源，请使用命令：

```
zypper repos
```

结果将类似于与以下输出：

#	Enabled	Refresh	Type	Alias	Name
1	Yes	Yes	yast2	openSUSE-DVD 11.0	openSUSE-DVD 11.0
2	Yes	No	yast2	Main (OSS)	Main (OSS)
3	Yes	No	yast2	Main (Non-OSS)	Main (Non-OSS)

如果想从此列表中删除某个安装源，请将命令 `zypper renamerepo` 和要删除的安装源的别名结合使用。要删除示例中的 Main Repository (Non-OSS)，请使用以下命令：

```
zypper renamerepo Main Repository (Non-OSS)
```

要添加安装源，请运行

```
zypper addrepo URI Alias
```

URI 可以是因特网安装源（请参见http://en.opensuse.org/Additional_YaST_Package_Repositories 以获取可用安装源列表）、目录、CD 或 DVD。*Alias* 是安装源的唯一简写标识符。您可以自由选择别名，唯一的例外情况是别名必须唯一。如果指定的别名已在使用，*zypper* 将发出警告。

4.1.5 查询

可以使用各种查询命令，如 *search*、*info* 或 *what-provides*。

search 会按包名进行操作，并会在输出的第一列中显示状态 (s) 信息。

带有作为自变量的包名的 *info* 会显示包的详细信息。

what-provides package 类似于 *rpm -q --whatprovides package*，但是 *rpm* 只能查询 RPM 数据库（即所有已安装包的数据库）。另一方面，*Zypper* 将告诉您任意安装源的功能的提供商，而非仅已安装的安装源功能的提供商。

特别是对于调试目的，可以使用如 *--plus-repo*、*--disable-repositories* 和 *--disable-system-resolvables* 之类的开关。如果只想在一个安装源中进行搜索，请使用这些开关。有关详细使用信息，请参见 *zypper* 手册页 (*man zypper*)。

4.1.6 使用 Zypper 壳层

有时必须依次运行几个不同的 *zypper* 命令。要避免 *zypper* 为每个 *zypper* 命令重复读取所有数据库，可以以壳层模式运行 *zypper*：

```
zypper shell
```

只需在运行壳层时使用相应的参数发出 *zypper* 命令：

```
zypper> in zsh
...
zypper> exit
```

使用 *zypper* 壳层的速度通常更快，因为所有的相关数据都在内存中。

Zypper 支持 readline 库。这意味着您可以使用 Zypper 壳层中的所有命令行编辑功能（这些功能在 Bash 壳层中也可用）。Zypper 在文件 `~/.zypper_history` 中维护其命令历史记录。

4.1.7 更多信息

有关从命令行更新的更多信息，请输入 `zypper --help` 或参见 `zypper(8)` 手册页。有关示例和详细信息，请访问 <http://en.opensuse.org/Zypper/Usage>。

4.2 RPM — 包管理器

RPM（RPM 程序包管理器）用于管理软件包。其主要命令为 `rpm` 和 `rpmbuild`。用户、系统管理员和包构建人员可以查询强大的 RPM 数据库以获得有关已安装软件的详细信息。

本质上，`rpm` 有五种模式：安装、卸装或更新软件包；重建 RPM 数据库；查询 RPM 库或独立 RPM 存档；包的完整性检查以及签署包。`rpmbuild` 可用于从原始源构建可安装的包。

用特殊的二进制格式对可安装 RPM 存档进行打包。这些存档由要安装的程序文件和某些元信息组成，这些元信息供 `rpm` 在安装过程中配置软件包使用或者储存在 RPM 数据库中进行存档。RPM 存档通常具有扩展名 `.rpm`。

提示: 软件开发包

对于许多包，已将软件开发所需的部件（库、标题、包含文件等）放入单独的包中。只有当您自己要自己编译软件时才需要这些开发包（例如最新的 GNOME 包）。可以通过名称扩展 `-devel` 确定这些开发包，例如包 `alsa-devel`、`gimp-devel` 和 `kdelibs3-devel`。

4.2.1 校验包真实性

RPM 包具有 GnuPG 签名。包括指纹的密钥是：

```
1024D/9C800ACA 2000-10-19 SuSE Package Signing Key <build@suse.de>  
Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA
```

命令 `rpm --checksig package-1.2.3.rpm` 可用于校验 **RPM** 包的签名，从而确定它确实来自 **SUSE** 还是其他可信工具。特别建议对来自因特网的更新包使用此命令。**SUSE** 公共包签名密钥通常驻留在 `/root/.gnupg/` 中。该密钥还位于目录 `/usr/lib/rpm/gnupg/` 中，允许一般用户校验 **RPM** 包的签名。

4.2.2 管理包：安装、更新和卸装

通常，**RPM** 存档的安装十分简单：`rpm -i package.rpm`。使用此命令可以安装包，但前提是满足其依赖性并且不与其他包冲突。如果出现错误消息，`rpm` 将请求那些需要安装的包以满足依赖性要求。在后台，**RPM** 数据库确保不出现冲突——一个特定文件只能属于一个包。通过选择不同的选项，您可以强制 `rpm` 忽略这些默认设置，但这只供专家用户使用。否则将影响系统的完整性并可能使系统无法更新。

选项 `-U`（即 `--upgrade`）和 `-F`（即 `--freshen`）可用于更新包。例如，`rpm -F package.rpm`。此命令将删除旧版本的文件并立即安装新文件。两个版本之间的差别是 `-U` 安装系统中以前不存在的包，但 `-F` 只更新以前安装的包。更新时，`rpm` 使用以下策略小心更新配置文件：

- 如果配置文件未被系统管理员更改，则 `rpm` 将安装适当文件的新版本。系统管理员无需执行任何操作。
- 如果更新前配置文件已被系统管理员更改，则 `rpm` 将以扩展名 `.rpmorig` 或 `.rpmsave`（备份文件）保存更改的文件并安装新包中的版本，但前提是原先安装的文件和较新的版本不同。如果是这种情况，则将备份文件（`.rpmorig` 或 `.rpmsave`）与新安装的文件进行比较，并在新文件中再次进行更改。随后，确保删除所有 `.rpmorig` 和 `.rpmsave` 文件以避免以后的更新出现问题。
- 如果配置文件已存在并且 `.spec` 文件中指定了 `noreplace` 标签，则出现 `.rpmnew` 文件。

更新后，在使用 `.rpmsave` 和 `.rpmnew` 文件进行比较后应将它们删除，从而防止它们阻碍以后的更新。如果 **RPM** 数据库以前未能识别文件，则将其指派扩展名 `.rpmorig`。

否则，将使用 `.rpmsave`。换句话说，`.rpmorig` 是从异系统格式更新为 **RPM** 的结果。而 `.rpmsave` 是从较早的 **RPM** 更新为较新的 **RPM** 的结果。`.rpmnew` 不提供任何有关系统管理员是否对配置文件进行了任何更改的信息。`/var/adm/rpmconfigcheck` 中提供这些文件的列表。不覆盖某些配置文件（如 `/etc/httpd/httpd.conf`）以允许继续进行操作。

`-U` 开关不仅仅是使用 `-e` 选项进行卸载并使用 `-i` 选项进行安装的等效项。只要可能，就可以使用 `-U`。

要删除包，请输入 `rpm -e package`。`rpm` 只在依赖性问题都解决的情况下才会删除该包。例如，只要有其他程序需要 `Tcl/Tk`，理论上就不能删除它。即使是在这种情况下，**RPM** 也会向数据库寻求帮助。如果出于任何原因或在任何特殊情况下不能进行这一删除操作（即使不存在任何其他依赖性），则最好使用 `--rebuilddb` 选项重建 **RPM** 数据库。

4.2.3 RPM 和增补程序

为了确保系统的操作安全性，必须时常在系统中安装更新包。以前，包中的 **bug** 只能通过替换整个包来解决。这样，对只有小文件中存在错误的较大的包进行替换时就很容易产生大量数据。不过 **SUSE RPM** 提供了一项功能，支持在包中安装增补程序。

以下使用 **pine** 的示例中对最重要的考虑事项进行了描述：

增补程序 **RPM** 是否适合我的系统？

要对此进行检查，请先查询包的已安装版本。对于 **pine**，可以通过以下命令完成：

```
rpm -q pine
pine-4.44-188
```

然后检查增补程序 **RPM** 是否适合此版本的 **pine**：

```
rpm -qp --basedon pine-4.44-224.i586.patch.rpm
pine = 4.44-188
pine = 4.44-195
pine = 4.44-207
```


此增补程序适用于 pine 的三个不同的版本。还列出示例中已安装的版本，从而可以安装增补程序。

增补程序将替换哪些文件？

在增补程序 RPM 中可以方便地找到受增补程序影响的文件。rpm 参数 -P 允许选择特殊的增补程序功能。使用以下命令显示文件列表：

```
rpm -qpPl pine-4.44-224.i586.patch.rpm
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

或者，如果已安装增补程序，则使用以下命令：

```
rpm -qPl pine
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

如何在系统中安装增补程序 RPM？

增补程序 RPM 的使用与普通 RPM 相同。唯一的区别就是必须已安装合适的 RPM。

系统中已安装了哪些增补程序，用于哪些包版本？

使用命令 rpm -qPa 可以显示系统中已安装的所有增补程序的列表。如果新系统中只安装了一个增补程序（如本示例中），则列表如下：

```
rpm -qPa
pine-4.44-224
```

如果以后要了解最初安装了哪个包版本，则可以在 RPM 数据库中获得此信息。对于 pine，可以通过以下命令显示此信息：

```
rpm -q --basedon pine
pine = 4.44-188
```

rpm 和 rpmbuild 的手册页中提供了详细信息（包括有关 RPM 的增补程序功能的信息）。

4.2.4 增量 RPM 包

增量 RPM 包包含旧版本和新版本的 RPM 包之间的差别。在旧 RPM 上应用增量 RPM 将得到全新的 RPM。不需要旧 RPM 的副本，因为增量 RPM 可以与已安装的 RPM 一起工作。增量 RPM 包的大小甚至比增补程序 RPM 小，这有利

于通过因特网传送更新包。缺点是，涉及增量 **RPM** 的更新操作与使用纯粹 **RPM** 或增补程序 **RPM** 进行更新的情况相比，占用的 CPU 周期要长得多。

`prepdeltarpm`、`writedeltarpm` 和 `applydeltarpm` 二进制文件是增量 **RPM** 套件（包 `deltarpm`）的一部分并帮助您创建和应用增量 **RPM** 包。使用以下命令，创建名为 `new.delta.rpm` 的增量 **RPM**。以下命令假设 `old.rpm` 和 `new.rpm` 是存在的：

```
prepdeltarpm -s seq -i info old.rpm > old.cpio
prepdeltarpm -f new.rpm > new.cpio
xdelta delta -0 old.cpio new.cpio delta
writedeltarpm new.rpm delta info new.delta.rpm
```

最后，删除临时工作文件 `old.cpio`、`new.cpio` 和 `delta`。

如果旧包已经安装，则使用 `applydeltarpm` 可以从文件系统重新构建新的 **RPM**：

```
applydeltarpm new.delta.rpm new.rpm
```

如果不访问文件系统而从旧 **RPM** 得到它，请使用 `-r` 选项：

```
applydeltarpm -r old.rpm new.delta.rpm new.rpm
```

关于技术详细信息，请参见 `/usr/share/doc/packages/deltarpm/README`。

4.2.5 RPM 查询

使用 `-q` 选项，`rpm` 将初始化查询，使查看 **RPM** 存档（通过添加选项 `-p`）并查询已安装包的 **RPM** 数据库成为可能。可以使用多个开关指定所需信息的类型。请参见表 4.1 “最重要的 **RPM** 查询选项” [30]。

表 4.1 最重要的 *RPM* 查询选项

<code>-i</code>	包信息
<code>-l</code>	文件列表
<code>-f FILE</code>	查询包含文件 <i>FILE</i> 的包（必须使用 <i>FILE</i> 指定完整路径）

<code>-s</code>	带有状态信息的文件列表（间接指定 <code>-l</code> ）
<code>-d</code>	仅列出文档文件（间接指定 <code>-l</code> ）
<code>-c</code>	仅列出配置文件（间接指定 <code>-l</code> ）
<code>--dump</code>	带有完整详细信息文件列表（将用于 <code>-l</code> 、 <code>-c</code> 或 <code>-d</code> ）
<code>--provides</code>	列出包中可被另一个包通过 <code>--requires</code> 请求的功能
<code>--requires, -R</code>	包需要的功能
<code>--scripts</code>	安装脚本（预安装、后安装、卸载）

例如，命令 `rpm -q -i wget` 显示 例 4.1 “`rpm -q -i wget`” [31] 中所示的信息。

例 4.1 `rpm -q -i wget`

```

Name           : wget                               Relocations: (not relocatable)
Version        : 1.9.1                             Vendor: SUSE LINUX AG,
Nuernberg, Germany
Release        : 50                                Build Date: Sat 02 Oct 2004
03:49:13 AM CEST
Install date: Mon 11 Oct 2004 10:24:56 AM CEST      Build Host: f53.suse.de
Group          : Productivity/Networking/Web/Utilities Source RPM:
wget-1.9.1-50.src.rpm
Size           : 1637514                             License: GPL
Signature      : DSA/SHA1, Sat 02 Oct 2004 03:59:56 AM CEST, 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.
[...]
```

只有当您指定带有完整路径的完整文件名时，选项 `-f` 才起作用。根据需要提供任意多个文件名。例如，以下命令

```
rpm -q -f /bin/rpm /usr/bin/wget
```

产生：

```
rpm-4.1.1-191
wget-1.9.1-50
```

如果只知道部分文件名，则可以使用壳层脚本，如例 4.2 “搜索包的脚本” [32] 所示。当运行所显示的脚本时，将部分文件名以参数的形式传递给该脚本。

例 4.2 搜索包的脚本

```
#!/bin/sh
for i in $(rpm -q -a -l | grep $1); do
    echo "\"$i\" is in package:"
    rpm -q -f $i
    echo ""
done
```

命令 `rpm -q --changelog rpm` 按照日期显示有关特定包更改信息的详细列表。此示例显示有关包 `rpm` 的信息。

借助已安装的 **RPM** 数据库，可以进行校验检查。使用 `-V`、`-y` 或 `--verify` 对其进行初始化。使用此选项，`rpm` 显示安装后已被更改的包中的所有文件。`rpm` 使用 8 个字符符号给出有关以下更改的一些提示：

表 4.2 RPM 校验选项

5	MD5 校验和
S	文件大小
L	符号链接
T	修改时间
D	主要和次要设备编号
U	拥有者
G	组
M	方式（权限和文件类型）

对于配置文件，将输出字母 c。例如，对于 `/etc/wgetrc` (`wget`) 的更改：

```
rpm -V wget
S.5....T c /etc/wgetrc
```

RPM 数据库的文件被放置在 `/var/lib/rpm` 中。如果分区 `/usr` 的大小为 1 GB，则此数据库可能会占用将近 30 MB，特别是在完全更新之后。如果数据库比预期大得多，则最好使用选项 `--rebuilddb` 重建数据库。在执行此操作之前，制作旧数据库的备份。`cron` 脚本 `cron.daily` 每天制作数据库的副本（用 `gzip` 打包）并将这些副本储存在 `/var/adm/backup/rpmdb` 中。副本的数目是由 `/etc/sysconfig/backup` 中的变量 `MAX_RPMD_DB_BACKUPS`（默认为 5）控制的。对于 1 GB 的 `/usr`，单个备份的大小大约为 1 MB。

4.2.6 安装和编译源包

所有源包都带有 `.src.rpm` 扩展名（源 RPM）。

提示

源包可以从安装媒体复制到硬盘并使用 YaST 解压缩。但是，在包管理器中它们不会被标记为已安装 ([i])。这是因为源包不是在 RPM 数据库中输入的。只有已安装的操作系统软件列在 RPM 数据库中。安装“源包时，只将源代码添加到系统中。”

以下目录必须可用于 `/usr/src/packages` 中的 `rpm` 和 `rpmbuild`（除非在诸如 `/etc/rpmsrc` 这样的文件中指定自定义设置）：

SOURCES

代表原始源（`.tar.bz2` 或 `.tar.gz` 文件等）和特定于发布版本的调整（多为 `.diff` 或 `.patch` 文件）

SPECS

代表 `.spec` 文件，类似于元 Makefile，该文件控制构建进程

BUILD

在此目录中解包、增补和编译所有源

RPMS

储存完整的二进制包的位置

SRPMS

这里是源 RPM

通过 YaST 安装源包时，所有必需的组件都安装在 `/usr/src/packages` 中：源和调整项在 SOURCES 中，相关 `.spec` 文件在 SPECS 中。

警告

不要对系统部件（`glibc`、`rpm`、`sysvinit` 等）进行试验，因为这将会影响系统的可操作性。

下面的示例使用 `wget.src.rpm` 包。在使用 YaST 安装此包之后，您应该具有与以下所列相似的文件：

```
/usr/src/packages/SOURCES/nops_doc.diff
/usr/src/packages/SOURCES/toplev_destdir.diff
/usr/src/packages/SOURCES/wget-1.9.1+ipvmisc.patch
/usr/src/packages/SOURCES/wget-1.9.1-brokentime.patch
/usr/src/packages/SOURCES/wget-1.9.1-passive_ftp.diff
/usr/src/packages/SOURCES/wget-LFS-20040909.tar.bz2
/usr/src/packages/SOURCES/wget-wrong_charset.patch
/usr/src/packages/SPECS/wget.spec
```

`rpmbuild -b X /usr/src/packages/SPECS/wget.spec` 启动编译。`X` 是通配符，代表构建进程的不同阶段（有关详细信息，请参见 `--help` 的输出或 RPM 文档）。以下内容只是简要描述：

`-bp`

在 `/usr/src/packages/BUILD` 中准备源：解压和打增补程序。

`-bc`

执行与 `-bp` 相同的操作，但还进行编译。

`-bi`

执行与 `-bp` 相同的操作，但还安装生成的软件。注意：如果包不支持 **BuildRoot** 功能，则可能会重写配置文件。

`-bb`

执行与 `-bi` 相同的操作，但还创建二进制包。如果编译成功，二进制包应该在 `/usr/src/packages/RPMS` 中。

-ba

执行与 -bb 相同的操作，但还创建源 RPM。如果编译成功，二进制包应该在 /usr/src/packages/SRPMS 中。

--short-circuit

跳过某些步骤。

现在可以使用 `rpm -i` 或最好使用 `rpm -U` 来安装创建的二进制 RPM。使用 `rpm` 进行安装使它显示在 RPM 数据库中。

4.2.7 使用 build 编译 RPM 包

许多包存在的风险是构建进程中会将许多不需要的文件添加到正在运行的系统中。为防止发生这种情况，请使用 `build`，它将创建构建包的确定环境。要建立这一 `chroot` 环境，`build` 脚本必须和完整的包树结构一起提供。可以通过 NFS 或从 DVD 使用硬盘上的此树。使用 `build --rpms directory` 设置位置。与 `rpm` 不同，`build` 命令在源目录中查找 SPEC 文件。要用系统中 /media/dvd 下装入的 DVD 构建 `wget`（如上面的示例中），请以 `root` 用户的身份使用以下命令：

```
cd /usr/src/packages/SOURCES/  
mv ../SPECS/wget.spec .  
build --rpms /media/dvd/suse/ wget.spec
```

随后，将在 /var/tmp/build-root 建立一个最小的环境。在此环境中构建包。完成后，生成的包位于 /var/tmp/build-root/usr/src/packages/RPMS 中。

`build` 脚本提供多个附加选项。例如，使脚本优先选择您自己的 RPM、忽略构建环境的初始化或者将 `rpm` 命令限制在上述阶段之一。使用 `build --help` 并通过阅读 `build` 手册页来访问更多信息。

4.2.8 用于 RPM 存档和 RPM 数据库的工具

Midnight Commander (`mc`) 可以显示 RPM 存档的内容并复制部分内容。它将存档表示为虚拟文件系统，提供 Midnight Commander 所有常用的菜单选项。使用 `F3` 键显示 HEADER。使用光标键和 `Enter` 键查看存档结构。使用 `F5` 键复制部分存档。

KDE 提供 `kpackage` 工具，作为 `rpm` 的前端。还提供作为 YaST 模块的功能齐全的包管理器（请参见第 7 章 *Installing or Removing Software* (↑部署指南)）。

通过 Nomad 访问远程桌面

Nomad (Novell Open Mobile Agile Desktop) 随 SUSE® Linux Enterprise Desktop 提供，它允许您运行与所有图形硬件分离的桌面会话。它包括以下核心组件：

代理 X 服务器

支持最新 X 扩展，如 Composite、XVideo 和 RANDR。

会话管理器

负责生成和跟踪可以远程访问的桌面会话。

连接处理程序

使用远程桌面协议 (RDP) 作为传输和安全层。RDP 是一个多通道协议，允许用户连接到运行 Microsoft Terminal Services 的计算机。但当客户端软件支持时，连接处理程序使用将未过滤的 X11 通讯传送到显示该桌面的本地 X 服务器的虚拟 X11 通道 (rdpx11)。连接处理程序始终可以回退到普通 RDP 命令（如果必要），这意味着可以从任何现有 RDP 客户端访问远程桌面。

客户端程序

为 SUSE Linux Enterprise Desktop 提供了一个特殊的 RDP 客户端，为 X11 协议转发实施特定于 Nomad 的扩展以及当装载相应的合成管理器插件时本地合成远程桌面的能力。

合成管理器扩展

合成有助于实现应用程序窗口的高级视觉效果，例如透明、渐弱、调整大小、扭曲、乱序和重定向。

Nomad 允许您从不同的物理位置远程访问桌面，例如您可以从家或办公室访问同一会话。工作会话中断后，只需移到另一个终端并继续工作。还可以将当前

运行的环境复制到移动设备（如便携式计算机）。有了 Nomad，还可以共享桌面以进行协作或培训，允许远程控制和管理。

5.1 Nomad 先决条件

为了使用 Nomad，需要在您的本地计算机上安装 `rdesktop` 包。此外，可以安装以下包：

- `compiz`
- `compiz-plugins-dmx`
- `compiz-fusion-plugins-main`
- `libcompizconfig`
- `python-compizconfig`
- `compiz-manager`
- `simple-ccsm`
- `tsclient`

在提供桌面的远程计算机上需要安装 `xrdp` 包，该包包含一个开放源代码远程桌面协议 (RDP) 服务器。

此外，可以安装以下包：

- `compiz`
- `compiz-plugins-dmx`
- `compiz-fusion-plugins-main`
- `libcompizconfig`
- `python-compizconfig`

- compiz-manager
- simple-ccsm

5.2 安装和设置

充当主机的本地计算机不需要任何特殊配置。安装 `rdesktop` 包后，即可以使用 `rdesktop` 命令行工具连接到提供桌面的远程计算机。如果要使用图形用户界面，请另外安装 `tsclient` 包。`tsclient`（终端服务器客户端）是 `rdesktop` 和其他远程桌面工具的 GNOME 前端，也支持 Xnest 和 VNC 客户端 (`vncviewer`)。要获得更佳性能和桌面效果，请另外安装 `compiz` 包。

但是，您需要如下准备提供桌面的远程计算机：

- 1 安装 `xrdp` 包。这将把 `xrpd` 服务器自动添加到运行级别 5。要手动启动或停止该服务，请作为 `root` 运行 `/etc/init.d/xrdp start` 或 `/etc/init.d/xrdp stop`。
- 2 配置防火墙以允许连接到端口 3389，因为此端口用于 RDP 连接。启动 YaST 并选择 **安全和用户 > 防火墙**。单击 **允许的服务** 并选择要允许该服务的区域。单击 **高级** 并输入 3389 作为 **TCP 端口**。在 YaST 中确认设置。
- 3 如果要使用 3D 桌面效果，请另外安装 `compiz` 包。当使用支持虚拟通道的客户端时，这将显著提高性能。通过在本地和远程桌面上同时启用桌面效果，本地组合管理器将能够对来自远程桌面的元素应用效果。

注意: 桌面效益

如果要在远程桌面上使用桌面效果，请确保在以下两个系统上都安装了 `compiz-plugins-dmx` 包：提供远程桌面的包的系统和访问远程桌面的本地系统。

5.3 使用 Nomad

在远程计算机上 `xrpd` 运行并且端口 3389 打开后，即可以使用您的 RDP 客户端连接到远程主机。使用 `rdesktop` 命令行工具或提供图形用户界面的 `tsclient` 连接。

5.3.1 使用 `rdesktop` 连接到服务器

要通过压缩模式对用户 `tux` 建立连接，请从外壳运行以下命令：

```
rdesktop -u tux -z server
```

其中，`server` 是远程计算机的主机名或 IP 地址。

这会为指定用户启动一个登录屏幕，该用户可以从此处登录到远程桌面。通过 `xrpd` 的桌面会话是独立的，与诸如 GDM 或 KDM 等普通显示管理器不冲突。

建立该连接时，您可以设置若干选项。例如，可以使用全屏模式，选择某种键盘布局或调整几何属性。通过 `rdesktop --help` 可了解更多有关可用 `rdesktop` 选项的信息。

5.4 查错

如果在建立连接时遇到困难，则根据以下列表继续操作。

在远程计算机上是否已启动并在运行 `xrpd` 服务器？

1. 检查提供桌面的远程计算机上是否安装了 `xrpd` 包。
2. 检查 `xrpd` 服务是否正在运行。
3. 否则，作为 `root` 执行以下命令而手动启动或重新启动它：
`/etc/init.d/xrpd start` 或 `/etc/init.d/xrpd restart`。

启动 `xrpd` 服务之后应有两个进程在运行：`xrpd` 和 `xrpd-sesman`。如果其中一个因为某种原因未能启动，则在前台手动启动这些进程将很可能告诉您出错的原因。

4. 要手动启动这些进程，请转换为 `root` 用户并运行
`/usr/sbin/xrdp-sesman -n` 和 `/usr/sbin/xrdp -nodaemon`。
5. 还要在 `/var/log/xrdp-sesman.log` 中检查 `xrdp-sesman` 输出，在
`/var/log/messages` 中检查 `xrdp` 输出，以获取更多信息。

5.5 更多信息

有关 Nomad 的详细信息，请参见 <http://en.opensuse.org/Nomad>。

Bash 和 Bash 脚本

今天，许多人使用具有诸如 KDE 或 GNOME 的图形用户界面 (GUI) 的计算机。尽管他们提供大量功能，当要执行自动任务时，它们的用途非常有限。外壳是对 GUI 的很好补充，本章提供关于外壳（在本案例中为 Bash）的某些方面的概述。

6.1 什么是“外壳”？

通常来说，外壳就是指 Bash（Bourne again 外壳）。当本章提到“外壳”时，指的是 Bash。实际上除了 Bash 之外，还有许多外壳，功能和特点都不同。如果需要关于其他外壳的更多信息，请在 YaST 中搜索外壳。

6.1.1 了解 Bash 配置文件

可以通过如下方式调用外壳：

1. 作为交互登录外壳。当登录计算机时需要使用此方式，即使用 `--login` 选项调用 Bash，或通过 SSH 登录到远程计算机时使用此方式。
2. 作为“普通”交互外壳。通常在启动 `xterm`、`konsole` 或类似工具时使用。
3. 作为非交互外壳。当在命令行调用外壳脚本时使用。

根据所用外壳的类型，会读取不同的配置文件。下表显示登录和非登录外壳的配置文件。

表 6.1 登录外壳的 *Bash* 配置文件

文件	描述
<code>/etc/profile</code>	不要修改此文件，否则在下一次更新时可能损坏您的修改！
<code>/etc/profile.local</code>	如果扩展 <code>/etc/profile</code> ，则使用此文件
<code>/etc/profile.d/</code>	包含特定程序的系统范围配置文件
<code>~/.profile</code>	在此处插入特定于用户的登录外壳配置

表 6.2 非登录外壳的 *Bash* 配置文件

<code>/etc/bash.bashrc</code>	不要修改此文件，否则在下一次更新时可能损坏您的修改！
<code>/etc/bash.bashrc.local</code>	使用此文件插入系统范围的修改（仅 Bash ）
<code>~/bashrc</code>	在此处插入特定于用户的配置

此外，**Bash** 还使用更多文件：

表 6.3 用于 *Bash* 的特殊文件

文件	描述
<code>~/.bash_history</code>	包含已键入的所有命令的列表
<code>~/.bash_logout</code>	注销时使用

6.1.2 目录结构

下表提供了 **Linux** 系统上最重要的较高级别目录的简短概述。以下列表中是关于这些目录和重要子目录的更多详细信息。

表 6.4 标准目录树概述

目录	内容
/	根目录 — 目录树的起点。
/bin	基本二进制文件，例如系统管理员和普通用户都需要的命令。通常还包含外壳，如 Bash 。
/boot	引导加载程序的静态文件。
/dev	访问特定于主机的设备所需的文件。
/etc	特定于主机的系统配置文件。
/home	储存系统上有帐户的所有用户的用户主目录。仅 root 的用户主目录不在 /home 而是在 /root 中。
/lib	基本共享库和内核模块。
/media	可卸媒体的安装点。
/mnt	临时装入文件系统的安装点。
/opt	附加应用程序软件包。
/root	超级用户 root 的用户主目录。
/sbin	基本系统二进制文件。
/srv	系统提供的服务的数据。
/tmp	临时文件。
/usr	具有只读数据的辅助层次结构。
/var	变量数据，如日志文件。

目录	内容
/windows	只在系统上同时安装了 Microsoft Windows* 和 Linux 时可用。包含 Windows 数据。

以下列表关于这些目录中有哪些文件和子目录提供更多详细信息，并给出一些示例：

/bin
包含 root 和其他用户都可使用的基本外壳命令。这些命令包含 ls、mkdir、cp、mv、rm 和 rmdir。/bin 也包含 Bash，它是 SUSE Linux Enterprise Desktop 中的默认外壳。

/boot
包含引导所需的数据，例如引导加载程序和内核开始执行用户模式程序之前使用的其他数据。

/dev
储存代表硬件组件的设备文件。

/etc
包含控制诸如 X Window 系统等程序操作的本地配置文件。/etc/init.d 子目录包含引导过程中执行的脚本。

/home/username
储存在系统中建立帐户的所有用户的私人数据。这里的文件只能由其所有者或系统管理员修改。默认情况下，电子邮件目录和个人桌面配置以隐藏文件和目录的方式位于此处。KDE 用户可分别在 .kde 或 .kde4 中查找其桌面的个人配置数据，GNOME 用户在 .gconf 中查找。

注意: 网络环境中的用户主目录

如果在网络环境中工作，则您的用户主目录可能映射到文件系统中除 /home 之外的其他目录中。

/lib
包含引导系统和运行 root 文件系统上的命令所需的基本共享库。共享库相当于 Windows 中的 DLL 文件。

`/media`

包含 CD-ROM、USB 记忆棒和数字相机（如果它们使用 USB）等可卸媒体的安装点。`/media` 通常包含除系统硬盘驱动器之外的各类驱动器。可卸媒体插入或连接到系统并装入之后，可以从此处访问该媒体。

`/mnt`

此目录提供临时装入的文件系统的安装点。`root` 可以在此处装入文件系统。

`/opt`

保留用于安装其他软件。在此处可以找到可选软件和较大附加程序包。KDE3 位于此处，而 KDE4 和 GNOME 现在已移到 `/usr`。

`/root`

`root` 用户的用户主目录。`root` 的个人数据位于此处。

`/sbin`

如 `s` 所表明的，该目录储存超级用户的实用程序。`/sbin` 包含除 `/bin` 中的二进制文件外引导、还原和恢复系统所需的其他二进制文件。

`/srv`

储存系统提供的服务（如 FTP 和 HTTP）的数据。

`/tmp`

此目录由需要临时储存文件的程序使用。

`/usr`

`/usr` 与用户无关，而是 UNIX 系统资源的缩写。`/usr` 中的数据是可以在符合文件系统层次结构标准 (FHS) 的各个主机之间共享的静态只读数据。此目录包含所有应用程序并建立文件系统辅助层次结构。KDE4 和 GNOME 也位于此处。`/usr` 储存有大量子目录，例如 `/usr/bin`、`/usr/sbin`、`/usr/local` 和 `/usr/share/doc`。

`/usr/bin`

包含一般可访问的程序。

`/usr/sbin`

包含为系统管理员保留的程序，例如维修功能。

`/usr/local`

在此目录中，系统管理员可以安装本地的独立于分发包的扩展。

`/usr/share/doc`

储存系统的各种文档文件和发行描述。在 `manual` 子目录中，有此手册的联机版本。如果安装了多种语言，则此目录可能包含这些手册不同语言的版本。

在 `packages` 下是系统上安装的软件包中包含的文档。对于每个包，都会创建一个子目录 `/usr/share/doc/packages/packagename`，经常用于储存该包的字数文件，有时储存示例、配置文件或附加脚本。

如果系统上安装了操作指南，`/usr/share/doc` 还会包含 `howto` 子目录，其中有与 Linux 软件的安装和操作相关的许多任务的附加文档。

`/var`

`/usr` 用于储存静态只读的数据，而 `/var` 用于在系统操作期间写入并成为变量数据的数据，例如日志文件或假脱机数据。例如，系统的日志文件在 `/var/log/messages` 中（仅 `root` 可访问）。

`/windows`

只在系统上同时安装了 Microsoft Windows 和 Linux 时可用。包含系统的 Windows 分区上可用的 Windows 数据。是否可以编辑此目录中的数据取决于 Windows 分区使用的文件系统。如果是 FAT32，则您可以打开和编辑此目录中的文件。但是，对于 NTFS 文件系统，从 Linux 仅可读取 Windows 文件，但不能修改它们。

6.2 编写外壳脚本

外壳脚本是执行所有类型任务的便捷方式：收集数据、在文本中搜索单词或短语以及许多其他有用的操作。以下示例显示用于打印文本的小外壳脚本：

例 6.1 用于打印文本的外壳脚本

```
#!/bin/sh ❶  
# Output the following line: ❷  
echo "Hello World" ❸
```

- ❶ 第一行开头是 *Shebang* 字符 (`#!`)，它表示该文件是一个脚本。该脚本使用 *Shebang* 后的指定解释程序执行，在本示例中为 `/bin/sh`。
- ❷ 第二行是一个以哈希符号开头的注释。建议对比较难理解的行进行注释以记住它的作用。
- ❸ 第三行使用内置命令 `echo` 打印相应文本。

可以运行该脚本之前，需要一些先决条件：

1. 每个脚本都应包含一个 *Shebang* 行（始终和我们的上述示例一样）。如果一个脚本没有该行，则必须自己调用解释程序。
2. 可以将该脚本保存在任何位置。但是，建议将其保存在外壳可以搜索得到的目录中。外壳中的搜索路径由环境变量 `PATH` 确定。例如，将其保存在 `~/bin/` 目录中并命名为 `hell .sh`。
3. 该脚本需要可执行权限。使用以下命令设置权限：

```
chmod +x ~/bin/hello.sh
```

如果已满足所有上述先决条件，则可以使用 `~/bin/hello.sh` 或 `hello.sh` 执行该脚本。第一个调用使用绝对路径，而第二个调用在 `PATH` 环境变量中提供的每个目录中搜索该命令。

6.3 重定向命令事件

每个命令都可以使用三个通道输入或输出：

- **标准输出** 这是默认的输出通道。在命令打印某些内容时都会使用标准输出通道。
- **标准输入** 如果一个命令需要用户或其他命令输入，则使用此通道。
- **标准错误** 命令使用此通道报告错误。

要重定向这些通道，有以下可行的操作方式：

命令 > 文件

将该命令的输出保存为文件，将删除现有文件。例如，`ls` 命令会将其输出写入文件 `listing.txt`：

```
ls > listing.txt
```

命令 >> 文件

将命令输出追加到文件。例如，`ls` 命令会将其输出追加到文件 `listing.txt`：

```
ls >> listing.txt
```

命令 < 文件

读取该文件作为给定命令的输入。例如，`read` 命令将该文件的内容读入一个变量：

```
read a < foo
```

命令 1 | 命令 2

将左侧命令的输出重定向为右侧命令的输入。

每个通道都有一个文件描述符：0 表示标准输入，1 表示标准输出，2 表示标准错误。允许在 < 或 > 字符前插入此文件描述符。例如，以下行搜索以 `foo` 开头的文件，但通过将错误重定向到 `/dev/null` 而抑制错误。

```
find / -name "foo*" 2>/dev/null
```

6.4 使用别名

别名是一个或多个命令的快捷方式定义。别名的语法为：

```
alias NAME=DEFINITION
```

例如，以下行定义了一个别名 `lt`，它输出一个长列表（选项 `-l`），将其按修改时间排序（`-t`），并在排序后倒序打印（`-r`）：

```
alias lt='ls -ltr'
```

要查看所有别名定义，请使用 `alias`。

6.5 在 Bash 中使用变量

外壳变量可以是全局变量，也可以是局部变量。全局变量（或环境变量）可以在所有外壳中访问。而局部变量仅在当前外壳中可见。

要查看所有环境变量，请使用 `printenv` 命令。如果需要特殊变量，请将该变量的名称作为自变量插入：

```
printenv PATH
```

变量也可以使用 `echo` 查看：

```
echo $PATH
```

这会打印 `PATH` 变量。要设置局部变量，请使用变量名后加等号和值：

```
PROJECT="SLED"
```

不要在等号两边插入空格，否则会出错。要设置环境变量，请使用 `export`：

```
export NAME="tux"
```

要删除变量，请使用 `unset`：

```
unset NAME
```

下表包含外壳脚本中可以使用的一些常见环境变量：

表 6.5 有用的环境变量

HOME	当前用户的用户主目录
HOST	当前主机名
LANG	当一个工具本地化后，它使用此环境变量中的语言。 英语也可以设置为 <code>C</code> 。
PATH	外壳的搜索路径，冒号分隔的目录列表。
PS1	指定在每个命令前打印的普通提示符
PS2	指定在执行多行命令时打印的辅助提示符

PWD

当前工作目录

USER

当前用户

6.5.1 使用自变量

例如，如果具有脚本 `foo.sh`，则可以如下执行：

```
foo.sh "Tux Penguin" 2000
```

要访问传递给脚本的所有这些自变量，您需要定位参数。`$1` 表示第一个自变量，`$2` 表示第二个自变量，依此类推。至多可以有九个参数。要获取脚本名称，请使用 `$0`。

以下脚本 `foo.sh` 打印从 1 到 4 的所有自变量：

```
#!/bin/sh
echo \"$1\" \"$2\" \"$3\" \"$4\"
```

如果使用上述自变量执行此脚本，将获取：

```
"Tux Penguin" "2000" "" ""
```

6.5.2 使用变量替换

变量替换将一个模式应用到变量的内容（从左侧或从右侧）。以下列表包含可能的语法格式：

`${VAR#pattern}`

从左侧删除可能的最短匹配：

```
file=/home/tux/book/book.tar.bz2
echo ${file#*/}
home/tux/book/book.tar.bz2
```

`${VAR##pattern}`

从左侧删除可能的最长匹配：

```
file=/home/tux/book/book.tar.bz2
echo ${file##*/}
book.tar.bz2
```



```
${VAR%pattern}
```

从右侧删除可能的最短匹配：

```
file=/home/tux/book/book.tar.bz2
echo ${file%.*}
/home/tux/book/book.tar
```

```
${VAR%%pattern}
```

从右侧删除可能的最长匹配：

```
file=/home/tux/book/book.tar.bz2
echo ${file%%.*}
/home/tux/book/book
```

6.6 将命令分组和组合

这些外壳允许您对命令连接和分组以有条件地执行。每个命令都返回一个退出码，该退出码确定操作是成功还是失败。如果是0，则命令成功，任何其他值都表示特定于该命令的一个错误。

以下列表显示可以如何将命令分组：

命令 1 ; 命令 2

顺序地执行这些命令。不检查退出码。以下行使用 `cat` 显示文件的内容，然后使用 `ls` 打印其文件属性，而不考虑退出码：

```
cat filelist.txt ; ls -l filelist.txt
```

命令 1 && 命令 2

如果左侧命令成功，则运行右侧命令（逻辑运算符 **AND**）。仅当上一个命令成功时，以下行才显示文件的内容并打印其文件属性（将其与列表中的上一项相比较）：

```
cat filelist.txt && ls -l filelist.txt
```

命令 1 || 命令 2

当左侧命令失败时运行右侧命令（逻辑运算符 **OR**）。以下行仅当在 `/home/tux/foo` 中创建目录失败时才会 `/home/wilber/bar` 中创建目录：

```
mkdir /home/tux/foo || mkdir /home/wilber/bar
```

```
funcname() { ... }
```

创建外壳函数。您可以使用定位参数访问其自变量。以下行定义用于打印短消息的函数 `hello`:

```
hello() { echo "Hello $1"; }
```

您可以如下调用此函数:

```
hello Tux
```

它会打印:

```
Hello Tux
```

6.7 使用通用流程构造语句

为了控制脚本的流程，脚本有 `while`、`if`、`for` 和 `case` 等构造语句。

6.7.1 if 控制命令

`if` 用于检查表达式。例如，以下代码测试当前用户是否是 `Tux`:

```
if test $USER = "tux" then
    echo "Hello Tux."
else
    echo "You are not Tux."
fi
```

测试表达式既可以复杂也可以简单。以下表达式检查文件 `foo.txt` 是否存在。

```
if test -e /tmp/foo.txt
then
    echo "Found foo.txt"
fi
```

请在 <http://www.cyberciti.biz/nixcraft/linux/docs/uniqlinuxfeatures/lsst/ch03sec02.html> 处查找更多表达式。

6.7.2 使用 for 命令创建循环

`for` 循环允许您对一系列项执行命令。例如，以下代码打印关于当前工作目录中 PNG 文件的某些信息:

```
for i in *.png; do
    ls -l $i
done
```

6.8 更多信息

关于 Bash 的重要信息在手册页 `man sh` 中提供。可以在以下列表中找到关于此主题的更多信息：

- <http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html> — Bash 入门者指南
- <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html> — BASH 编程 - 简介操作指南
- <http://tldp.org/LDP/abs/html/index.html> — 高级 Bash 脚本编写指南
- <http://www.grymoire.com/Unix/Sh.html> — sh - Bourne 外壳

部分 II. 系统

64 位系统环境中的 32 位和 64 位应用程序

SUSE® Linux Enterprise Desktop 可用于 64 位平台。但是这并不表示内含的所有应用程序都已移植到 64 位平台上。SUSE Linux Enterprise Desktop 支持在 64 位系统环境中使用 32 位应用程序。本章简单介绍了如何在 64 位 SUSE Linux Enterprise Desktop 平台上实现这种支持。它解释了如何执行 32 位应用程序（运行时支持）以及应该如何编译 32 位应用程序以使它们既可以在 32 位系统环境中运行，又可以在 64 位系统环境中运行。另外，您还可以了解有关内核 API 的信息和 32 位应用程序如何在 64 位内核下运行的解释。

用于 64 位平台 amd64 和 Intel 64 的 SUSE Linux Enterprise Desktop 被设计为可以让现有的 32 位应用程序可以在 64 位环境下运行，而“无需进行额外设置。”这种支持意味着您可以继续使用所需的 32 位应用程序，而无需等待对应的 64 位端口可用。

7.1 运行时支持

重要：应用程序版本之间的冲突

如果某个应用程序在 32 位和 64 位环境中都可用，则两个版本的并行安装必定会导致出现问题。在这种情况下，在两个版本中选一个，然后安装并使用这一版本。

该规则的一个例外是 PAM（可插入身份验证模块）。SUSE Linux Enterprise Desktop 在身份验证过程中使用 PAM 作为在用户和应用程序之间充当媒介的层。在仍运行 32 位应用程序的 64 位操作系统上，需要始终安装 PAM 模块的两个版本。

若要正确执行，每个应用程序都需要一系列库。不巧的是，这些库的 32 位和 64 位版本的名称是相同的。必须通过另一种方法对它们加以区分。

为了保持与 32 位版本的兼容性，这些库在系统中的储存位置与在 32 位环境中相同。在 32 位和 64 位环境中，`libc.so.6` 的 32 位版本都位于 `/lib/libc.so.6` 下。

所有 64 位库和对象文件都位于名为 `lib64` 的目录中。通常可以在 `/lib` 下找到 64 位对象文件，现在可以在 `/lib64` 和 `/usr/lib64` 下找到 `/usr/lib`。这意味着 `/lib` 和 `/usr/lib` 下有储存 32 位库的空间，因此两个版本的文件名都可以保持不变。

如果对象目录的数据内容不取决于此大小，则不移动 32 位 `/lib` 目录的任何子目录。此方案符合 LSB（Linux 标准库）和 FHS（文件系统层次标准）。

7.2 软件开发

`biarch` 开发工具链允许生成 32 位和 64 位对象。默认为编译 64 位对象。通过使用特殊标志也可以生成 32 位对象。对于 GCC，此特殊标志是 `-m32`。

必须以一种独立于体系结构的形式编写所有头文件。安装的 32 位和 64 位库必须具有与安装的头文件匹配的 API（应用程序编程接口）。常规 SUSE Linux Enterprise Desktop 环境是根据此原则设计的。如果是手动更新的库，请自行解决此问题。

7.3 Biarch 平台上的软件编译

若要在 Biarch 体系结构上为其他体系结构开发二进制代码，则必须另外安装用于第二个体系结构的各个库。这些包称为 `rpmname-32bit`。您还需要 `rpmname-devel` 包中各自的标题和库以及 `rpmname-devel-32bit` 中用于第二个体系结构的开发库。

大多数开放源代码程序使用基于 `autoconf` 的程序配置。若要使用 `autoconf` 配置第二个体系结构的程序，请通过运行带有附加环境变量的 `configure` 脚本覆盖 `autoconf` 的常规编译器和链接器设置。

以下示例涉及 x86 为第二个体系结构的 x86_64 系统。

1 使用 32 位编译器：

```
CC="gcc -m32"
```

2 指示链接器处理 32 位对象（总是使用 gcc 作为链接器前端）：

```
LD="gcc -m32"
```

3 设置组装机生成 32 位对象：

```
AS="gcc -c -m32"
```

4 确定 libtool 等的库是否来自 /usr/lib：

```
LDFLAGS="-L/usr/lib"
```

5 确定库是否储存在 lib 子目录中：

```
--libdir=/usr/lib
```

6 确定是否使用了 32 位 X 库：

```
--x-libraries=/usr/lib/xorg
```

并不是每个程序都需要这些变量。根据各个程序对这些变量进行调整。

```
CC="gcc -m32" \
LDFLAGS="-L/usr/lib;" \
    .configure \
        --prefix=/usr \
        --libdir=/usr/lib
make
make install
```

7.4 内核规范

x86_64 的 64 位内核提供了 64 位和 32 位内核 ABI（应用程序二进制接口）。后者与对应的 32 位内核的 ABI 相同。这意味着 32 位应用程序可以以与 32 位内核交流的相同方式与 64 位内核进行交流。

64 位内核系统调用的 32 位仿真不支持系统程序使用的某些 API。这取决于平台。因此，必须编译少量的应用程序，如 `lspci`

64 位内核只能装载专门为此内核编译的 64 位内核模块。不能使用 32 位内核模块。

提示

某些应用程序需要单独的内核可装载模块。如果要在 64 位系统环境中使用此类 32 位应用程序，请与此应用程序的提供商和 Novell 联系以确保内核可装载模块的 64 位版本和内核 API 的 32 位编译版本可用于此模块。

引导和配置 Linux 系统

引导 Linux 系统包括多个不同组件。硬件本身是由 BIOS 初始化，而 BIOS 通过引导加载程序启动内核。此后，引导过程（包括 `init` 和 `runlevel`）完全受操作系统控制。`runlevel` 概念使您可以维护日常使用的设置，也可以对系统执行维护任务。

8.1 Linux 引导进程

Linux 引导进程包括多个阶段，每个阶段由一个组件来代表。下表概要总结了引导进程并介绍了所涉及的所有主要组件。

1. **BIOS** 在打开计算机后，BIOS 将初始化屏幕和键盘并测试主存储器。直到这一阶段，计算机不访问任何大容量储存媒体。随后，将从 CMOS 值装载有关当前日期、时间和最重要的外设的信息。当识别出第一块硬盘及其空间之后，系统控制将从 BIOS 传递到引导加载程序。
2. **Boot Loader** 第一块硬盘的前 512 个字节的物理数据扇区将被装载到主存储器中，位于此扇区开始位置的引导加载程序将接管系统控制。引导加载程序执行的命令决定了引导进程剩余的部分。因此，第一块硬盘的前 512 个字节被称为主引导记录 (MBR)。引导加载程序随后将控制传递到实际的操作系统（在本例中即 Linux 内核）。有关 Linux 引导加载程序 GRUB 的详细信息，请参见第 9 章 *引导加载程序 GRUB* [77]。
3. **内核和 `initramfs`** 为了传递系统控制，引导加载程序将内核和基于 RAM 的初始文件系统 (`initramfs`) 装载到内存中。内核可以直接使用 `initramfs` 的内容。`initramfs` 包含一个小的可执行文件，称为 `init`，可以进行真实文件系

统的装入处理。如果在访问大容量储存区之前需要特殊硬盘驱动程序，则这些程序必须在 `initramfs` 中。有关 `initramfs` 的详细信息，请参见第 8.1.1 节“`initramfs`” [64]。

4. **initramfs 中的 `init`** 这个程序执行装入正确的 `root` 文件系统所需的所有操作，如为所需的文件系统提供内核功能以及为带有 `udev` 的大容量储存控制器提供设备驱动程序。找到 `root` 文件系统后，对其进行错误检查并装入。如果该操作成功，将清除 `initramfs` 并执行 `root` 文件系统上的 `init` 程序。有关 `init` 的详细信息，请参见第 8.1.2 节“`initramfs 中的 init`” [65]。有关 `udev` 的更多信息，请参见第 12 章 *使用 `udev` 进行动态内核设备管理* [123]。
5. **`init`** `init` 通过提供不同功能的多个不同的级别来处理系统的实际引导。有关 `init` 的介绍，请参见第 8.2 节“`init 进程`” [66]。

8.1.1 `initramfs`

`initramfs` 是一个小型 `cpio` 归档，在此内核可以装载到 `RAM` 磁盘。它提供了一个最小的 `Linux` 环境，可在装入实际 `root` 文件系统之前执行程序。这个最小的 `Linux` 环境由 `BIOS` 例程装载进内存，而且除了需要足够的内存外没有特别的硬件要求。`initramfs` 必须始终提供一个名为 `init` 的可执行文件，该文件应该执行 `root` 文件系统中实际的 `init` 程序以使引导进程继续进行。

在能够装入 `root` 文件系统并启动操作系统之前，内核需要相应的驱动程序来访问 `root` 文件系统所在的设备。这些驱动程序可能包括用于特定类型硬盘的特殊驱动程序，甚至还可能包括访问网络文件系统所需的网络驱动程序。可使用 `initramfs` 上的 `init` 装载根目录文件系统所需的模块。模块装载之后，`udev` 将为 `initramfs` 提供所需的设备。在引导过程的后面，更改 `root` 文件系统之后需要重新生成设备。通过 `boot.udev`（使用 `udevtrigger` 命令）来完成此操作。

如果需要在已安装的系统中更改硬件（例如硬盘），并且该硬件要求在引导时内核中有不同的驱动程序，则必须更新 `initramfs`。其操作方法和其前身 `initrd` 一样，即调用 `mkinitrd`。调用 `mkinitrd` 无需任何参数便可创建 `initramfs`。调用 `mkinitrd -R` 创建 `initrd`。在 `SUSE® Linux Enterprise Desktop` 中，要装载的模块由 `/etc/sysconfig/kernel` 中的变量 `INITRD_MODULES` 指定。安装后，自动将此变量设置为正确的值。将严格按照这些模块在 `INITRD_MODULES` 中出现的顺序来装载它们。只有您依赖正确的设备文件 `/dev/sd?` 设置时，这才显得重要。然而，在当前系统下，也可以使用 `/dev/disk/` 下的设备文件。

这些文件以几个子目录的形式排序，分别为 `by-id`、`by-path` 和 `by-uuid`，并始终代表相同的磁盘。也可以在安装时通过指定相应的装入选项完成此操作。

重要: 更新 `initramfs` 或 `initrd`

引导加载程序装载 `initramfs` 或 `initrd` 的方式与内核相同。在更新 `initramfs` 或 `initrd` 后无需重安装 GRUB，因为 GRUB 会在引导时搜索目录以获得正确的文件。

8.1.2 `initramfs` 中的 `init`

`initramfs` 中的 `init` 的主要用途是准备真实 `root` 文件系统的装入和访问。根据系统配置的不同，`init` 负责以下任务。

装载内核模块

根据硬件配置的不同，可能需要一些特殊的驱动程序来访问计算机的硬件组件（最重要的组件是硬盘）。要访问最终的 `root` 文件系统，内核需要装载适当的文件系统驱动程序。

提供块特殊文件

内核对每个装载的模块生成设备事件。`udev` 处理这些事件，并在 RAM 文件系统上的 `/dev` 中生成必需的块特殊文件。没有这些特殊文件，文件系统和其他设备将不可访问。

管理 RAID 和 LVM 设置

如果将系统配置为在 RAID 或 LVM 下保存 `root` 文件系统，则 `init` 将设置 LVM 或 RAID 以支持稍后对 `root` 文件系统的访问。在第 14 章 *Advanced Disk Setup* (↑部署指南) 中查找关于 RAID 和 LVM 的信息。

管理网络配置

如果对系统进行配置以使用通过网络装入的 `root` 文件系统（通过 NFS 装入），则 `init` 必须确保装载了适当的网络驱动器，并确保对其进行设置以允许对 `root` 文件系统的访问。

在初始引导期间调用 `init` 时（安装进程一部分），要执行的任务将与前面提到的任务不同：

查找安装媒体

启动安装进程时，计算机将通过安装媒体中的 YaST 安装程序装载一个安装内核和一个特殊的 `initrd`。YaST 安装程序在 RAM 文件系统中运行，它需要有关安装媒体位置的信息以访问安装媒体并安装操作系统。

启动硬件识别并装载适当的内核模块

如第 8.1.1 节“`initramfs`”[64]中所述，引导进程从可用于大多数硬件配置的一组最小的驱动程序启动。`init` 将启动初始硬件扫描进程，以确定适合您的硬件配置的一组驱动程序。引导进程所需的模块名写进

`/etc/sysconfig/kernel` 中的 `INITRD_MODULES`。这些名称用来生成引导该系统所需要的自定义 `initramfs`。如果模块不是用于引导，而是用于冷插入，则模块要写进 `/etc/sysconfig/hardware/hwconfig-*`。本目录下用配置文件描述的所有设备均要在引导过程中进行初始化。

装载安装系统或救援系统

一旦正确地识别出硬件并装载了适当的驱动程序并且 `udev` 创建了设备特殊文件后，`init` 就会启动安装系统，其中包含实际的 YaST 安装程序或救援系统。

启动 YaST

最后，`init` 将启动 YaST，由后者启动包安装和系统配置。

8.2 `init` 进程

`init` 程序是进程 ID 为 1 的进程，负责按所要求的方式对系统进行初始化。`init` 由内核直接启动，并且抵制信号 9（该信号通常会杀死进程）。所有其他程序由 `init` 直接启动，或由它的其中一个子进程启动。

`init` 在 `/etc/inittab` 文件中进行集中配置，其中运行级别已定义（请参见第 8.2.1 节“运行级别”[67]）。该文件还指定了在每个运行级别有哪些服务和守护程序可用。根据 `/etc/inittab` 中的项，`init` 将运行若干个脚本。默认情况下，引导后启动的第一个脚本为 `/etc/init.d/boot`。完成系统初始化阶段后，系统将使用 `/etc/init.d/rc` 脚本把运行级别更改为默认运行级别。为了清楚起见，这些称作 *init* 脚本的脚本都位于目录 `/etc/init.d` 中（请参见第 8.2.2 节“Init 脚本”[69]）。

启动和关闭系统的整个过程是由 `init` 维护的。从这一点来看，可以将内核视为一个后台进程，其任务是维护所有其他进程，以及根据其他程序的请求来调整 CPU 时间和硬件访问。

8.2.1 运行级别

在 Linux 中，运行级别定义了系统如何启动以及正在运行的系统中有哪些服务可用。在引导后，系统会按照 `/etc/inittab` 中的 `initdefault` 行所定义的方式启动。通常是 3 或 5。请参见表 8.1 “可用运行级别” [67]。也可以选择 在引导时指定运行级别（例如，在引导提示符后添加运行级别号）。任何不直接由内核本身求值的参数均将被传递给 `init`。要引导进运行级别 3 中，只需将单个数字 3 添加到引导提示中。

表 8.1 可用运行级别

运行级别	描述
0	系统暂停
S 或 1	单用户方式
2	没有远程网络的本地多用户方式（NFS 等）
3	有网络的完全多用户方式
4	除非管理员配置该运行级别，否则不使用用户定义。
5	有网络和 X 显示管理器的完全多用户方式 — KDM、GDM 或 XDM
6	系统重引导

重要: 避免运行级别 2 与通过 NFS 装入的分区

如果您的系统通过 NFS 装入了 `/usr` 分区，则不应使用运行级别 2。如果程序文件或库丢失，系统可能会异常运行，因为 NFS 设备不能以运行级别 2（没有远程网络的本地多用户方式）提供。

要在系统运行时更改运行级别，请输入 `telinit` 和作为参数的相应数字。仅允许系统管理员执行该操作。下表总结了运行级别区域中最重要的命令。

`telinit 1` 或 `shutdown now`

系统更改为单用户方式。该方式用于系统维护和管理任务。

`telinit 3`

启动了所有基本的程序和服务（包括网络），允许普通用户登录并在不具备图形环境的系统中工作。

`telinit 5`

启用了图形化环境。通常启动诸如 `XDM`、`GDM` 或 `KDM` 之类的显示管理器。如果启用 `autologin`，则本地用户便可登录到预先选择的窗口管理器（`GNOME` 或 `KDE` 或其他任何窗口管理器）中。

`telinit 0` 或 `shutdown -h now`

系统暂停。

`telinit 6` 或 `shutdown -r now`

系统暂停后重引导。

运行级别 5 是所有 SUSE Linux Enterprise Desktop 标准安装中的默认运行级别。提示用户使用图形界面登录，或者默认用户将自动登录。如果默认运行级别是 3，必须按照 [第 13 章 X Window 系统](#) [135] 中的描述正确配置 X Window 系统，才能将运行级别切换为 5。完成切换后，请通过输入 `telinit 5` 来检查系统是否以预期方式运行。如果一切合乎预期，就可以使用 YaST 将默认运行级别设置为 5。

警告: `/etc/inittab` 中的错误可能导致系统引导出现问题

如果 `/etc/inittab` 损坏，则可能无法正常引导系统。因此，在编辑 `/etc/inittab` 时要特别小心。在重引导计算机前，使 `init` 使用 `telinit q` 命令重读 `/etc/inittab`。

通常情况下，更改运行级别时会发生两件事情。首先是启动当前运行级别的停止脚本，同时关闭当前运行级别必需的一些程序。然后启动新运行级别的启动脚本。在大多数情况下，这时会启动多个程序。例如，将运行级别从 3 更改到 5 时会发生以下情况：

1. 通过输入 `telinit 5`，管理员 (root) 要求 `init` 更改为另一个运行级别。
2. `init` 检查当前运行级别 (`runlevel`) 并确定是否应使用新的运行级别作为参数来启动 `/etc/init.d/rc`。
3. `rc` 现在调用当前运行级别的停止脚本，但仅限新运行级别中没有启动脚本的那些停止脚本。在本例中，这些就是位于 `/etc/init.d/rc3.d` (旧的运行级别是 3) 中以 `K` 开头的所有脚本。`K` 后跟的编号指定使用 `stop` 参数运行脚本的顺序，因为有很多依赖性要考虑。
4. 最后要启动的是新运行级别的启动脚本。在本例中，这些是位于 `/etc/init.d/rc5.d` 中以 `S` 开头的脚本。`S` 后跟的编号确定启动脚本的顺序。

当更改为与当前运行级别相同的运行级别时，`init` 仅检查 `/etc/inittab` 的更改，并启动相应的步骤（例如，在另一个界面上启动 `getty` 所需的步骤）。使用命令 `telinit q` 也达到到相同的作用。

8.2.2 Init 脚本

`/etc/init.d` 中有两种类型的脚本：

由 `init` 直接执行的脚本

仅在引导过程中或在启动系统立即关闭时（电源故障或用户按了 **Ctrl + Alt + Del** 组合键）时才会发生这种情况。这些脚本的执行是在 `/etc/inittab` 中定义的。

由 `init` 间接执行的脚本

这些脚本在更改运行级别时运行并始终调用主脚本 `/etc/init.d/rc`，后者能够确保相关脚本以正确顺序运行。

所有脚本位于 `/etc/init.d` 中。引导时运行的脚本是通过指向 `/etc/init.d/boot.d` 的符号链接调用的。用于更改运行级别的脚本也是通过符号链接从一个子目录 (`/etc/init.d/rc0.d` 到 `/etc/init.d/rc6.d`) 进行调用的。这仅仅是为了清楚起见，并避免在多个运行级别使用时出现重复脚本。因为每个脚本既可以作为启动脚本也可以作为停止脚本来执行，这些脚本必须理解 `start` 和 `stop` 参数。这些脚本还必须理解 `restart`、`reload`、`force-reload` 和 `status` 选项。对这些不同的选项进行了解释。[表 8.2 “可](#)

能的 **init 脚本选项**” [70] 由 `init` 直接运行的脚本没有这些链接。需要时，可以从运行级别独立运行它们。

表 8.2 可能的 `init` 脚本选项

选项	描述
<code>start</code>	启动服务。
<code>stop</code>	停止服务。
<code>restart</code>	如果服务正在运行，则首先将其停止，然后重新启动。如果服务未在运行，则启动服务。
<code>reload</code>	在不停止和重新启动服务的情况下重装载配置。
<code>force-reload</code>	如果服务支持，则重装载配置。否则，要执行的步骤与指定 <code>restart</code> 时相同。
<code>status</code>	显示服务的当前状态。

每个特定于运行级别的子目录中的链接使将脚本与不同的运行级别相关联成为可能。在安装或卸载包时，在程序 `insserv`（或使用 `/usr/lib/lsb/install_initd`，它是调用此程序的一个脚本）的帮助下可添加和删除这些链接。有关详细信息，请参见手册页 `insserv(8)`。

所有这些设置也可能在 `YaST` 模块的帮助下发生变化。如果需要检查命令行的状态，请使用 `chkconfig(8)` 手册页中所描述的 `chkconfig` 工具。

下面分别简要介绍最先或最后启动的引导和停止脚本，并对脚本的维护进行了描述。

引导

在使用 `init` 直接启动系统时执行。它与选择的运行级别无关，而且仅执行一次。这时将装入 `/proc` 和 `/dev/pts` 文件系统，并激活 `blogd`（引导日志记录守护程序）。如果在更新或安装后首次引导系统，则会启动初始系统配置。

blogd 守护程序是由 **boot** 和 **rc** 启动的第一个服务。在由这些脚本触发的操作（运行几个子脚本，例如使块特殊文件变为可用的）完成之后它停止。**blogd** 将所有屏幕输出写入日志文件 `/var/log/boot.msg`（前提是装入的 `/var` 是可读写的）。否则，**blogd** 将缓冲所有屏幕数据，直到 `/var` 可用。有关 **blogd** 的详细信息，请参见手册页 **blogd(8)**。

root 脚本还负责启动 `/etc/init.d/boot.d` 中名称以 **s** 开头的脚本。在这里，将检查文件系统并根据需要配置回路设备。同时设置系统时间。如果在自动检查和修复文件系统时出错，系统管理员可以在输入根密码后进行干预。上次执行的脚本是 `boot.local`。

boot.local

在这里，输入引导时在更改为某个运行级别之前执行的其他命令。这类似于 **DOS** 系统上的 `AUTOEXEC.BAT`。

halt

仅当更改为运行级别 **0** 或 **6** 时执行该脚本。它在这里作为 **halt** 或 **reboot** 来执行。是关闭还是重引导系统取决于调用 **halt** 的方式。如果在关闭系统期间需要特殊命令，请将这些命令添加到 `halt.local` 脚本。

rc

此脚本调用当前运行级别的相应停止脚本和新选择的运行级别的启动脚本。与 `/etc/init.d/boot` 脚本类似，该脚本是通过将所需运行级别用作参数从 `/etc/inittab` 调用的。

您可以创建自己的脚本并方便地将它们集成到上面描述的方案中。有关格式化、命名和组织自定义脚本的描述，请参考 **LSB** 的规范以及 `init`、`init.d`、`chkconfig` 和 `insserv` 的手册页。此外还可以参见 `startproc` 和 `killproc` 的手册页。

警告: 有问题的 **init** 脚本可能会使您的系统暂停

有问题的 **init** 脚本可能会使您的计算机挂起。应认真编辑这些脚本，如果可能，应在多用户环境中对它们进行严格测试。第 8.2.1 节“运行级别”[67]中有一些有关 **init** 脚本的有用信息。

要为给定程序或服务创建自定义 **init** 脚本，请使用文件 `/etc/init.d/skeleton` 作为模板。以新名称保存此文件的备份，然后根据需要编辑相关程

序和文件名、路径及其他详细信息。您可能还需要用自己的部分来增强此脚本，以便 `init` 过程可以触发正确的操作。

位于顶部的 `INIT INFO` 块是脚本的一个必需部分，应进行编辑。请参见例 8.1 “最小的 `INIT INFO` 块” [72]。

例 8.1 最小的 `INIT INFO` 块

```
### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Required-Stop:     $syslog $remote_fs
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Description:       Start FOO to allow XY and provide YZ
### END INIT INFO
```

在 `INFO` 块第一行上 `Provides:` 后，指定由此 `init` 脚本控制的程序或服务的名称。在 `Required-Start:` 和 `Required-Stop:` 行中，指定在启动或停止服务本身之前，需要启动或停止的所有服务。这些信息稍后用于生成脚本名的编号（可以在运行级别目录中找到）。在 `Default-Start:` 和 `Default-Stop:` 后，指定应自动启动或停止的服务所在的运行级别。最后，在 `Description:` 下，提供对相关服务的简短描述。

要创建从运行级别目录 (`/etc/init.d/rc?.d/`) 到 `/etc/init.d/` 中相应脚本的链接，请输入命令 `insserv new-script-name`。`insserv` 程序对 `INIT INFO` 标题进行求值，以便为运行级别目录 (`/etc/init.d/rc?.d/`) 中的启动和停止脚本创建必要的链接。此程序还负责保证每个运行级别的启动和停止顺序正确无误，方法是在这些链接的名称中包含必要的数字。如果要使用图形工具来创建这样的链接，请按照第 8.2.3 节 “使用 **YaST 配置系统服务（运行级别）**” [73] 中描述的方法使用 **YaST** 提供的运行级别编辑器。

如果应将已存在于 `/etc/init.d/` 中的脚本集成到现有运行级别方案中，请立即通过 `insserv` 或启用 **YaST** 的运行级别编辑器中的相应服务在运行级别目录中创建链接。您的更改将在下次重引导时生效 — 新服务将自动启动。

不要手动设置这些链接。如果 `INFO` 块中出错，则在稍后为其他服务运行 `insserv` 时将会出现问题。下次为此脚本运行 `insserv` 时将删除手动添加的服务。

8.2.3 使用 YaST 配置系统服务（运行级别）

使用 *YaST* > 系统 > 系统服务（运行级别）启动此 YaST 模块后，它将显示一个概要，列出所有可用的服务和每个服务的当前状态（禁用或启用）。确定是以简单方式还是以专家方式使用此模块。默认的方式足以完成大多数操作。左边的列显示服务的名称，中间的列指示其当前状态，而右边的列则给出简短描述。窗口下部提供了对所选服务的更为详细的描述。若要启用某个服务，请首先在表中选定它，然后选择启用。同样的步骤可用于禁用服务。

要对所启动或停止的服务所在运行级别进行更具体的控制，或者更改默认运行级别，请先选择专家方式。将在顶部显示当前默认的运行级别或“initdefault”（默认情况下将系统引导至的运行级别）。通常情况下，SUSE Linux Enterprise Desktop 系统的默认运行级别是运行级别 5（有网络和 X 的完全多用户模式）。运行级别 3（有网络的完全多用户方式）是合适的替代选择。

此 YaST 对话框用于选择一个运行级别（如 [表 8.1 “可用运行级别”](#) [67] 中所列）作为新的默认运行级别。此外，可使用此窗口中的表来启用或禁用各个服务和守护程序。此表列出可用的服务和守护程序，显示它们当前是否已在您的系统上启用，如果已启用，则指示它们用于哪些运行级别。用鼠标选择其中的一行后，请单击表示运行级别（B、0、1、2、3、5、6 和 S）的复选框来确定所选服务或守护程序的运行级别。未对运行级别 4 进行定义，目的是供用户创建自定义运行级别。表概要下方提供了当前所选服务或守护程序的简要描述。

警告：有问题的运行级别设置可能会对您的系统造成损害

有问题的运行级别设置可能会导致系统无法使用。在应用您的更改之前，请确保您清楚这些设置可能产生的结果。

图 8.1 系统服务（运行级别）



用启动、停止或刷新来确定是否应激活某服务。刷新状态用来检查当前状态。设置或重设置用于选择是将更改应用到系统，还是恢复启动运行级别编辑器之前存在的设置。选择完成即可将已更改的设置保存到磁盘。

8.3 通过 /etc/sysconfig 配置系统

SUSE Linux Enterprise Desktop 的主要配置是由 /etc/sysconfig 中的配置文件控制的。只有与 /etc/sysconfig 中的各个文件相关的脚本才会读取它们。这样有很多好处，例如确保了网络设置只需要由与网络相关的脚本来分析。

可以使用两种方法编辑系统配置。使用 YaST sysconfig 编辑器或手动编辑配置文件。

8.3.1 使用 YaST Sysconfig 编辑器更改系统配置

YaST sysconfig 编辑器为系统配置提供了一种使用方便的前端。无需了解需要更改的配置变量的实际位置，只需使用该模块的内置搜索功能，就可以根据需要更改配置变量的值，并使 YaST 负责应用这些更改以及根据 sysconfig 中设置的值更新配置和重新启动服务。

警告: 修改 `/etc/sysconfig/*` 文件可能会对您的安装造成损害

如果没有足够的经验和知识，切勿修改 `/etc/sysconfig` 文件。否则可能会对您的系统造成巨大损害。`/etc/sysconfig` 中的文件包含对每个变量的简短注释，解释了这些变量的实际作用。

图 8.2 使用 sysconfig 编辑器进行系统配置



YaST sysconfig 对话框分为三个部分。对话框左边的部分显示了一个树视图，其中列出了所有可配置变量。当您选择某个变量时，右边的部分会显示当前选择和此变量的当前设置。在下部的第三个窗口中，简要描述了变量的用途、可能的值、默认值以及作为此变量来源的实际配置文件。此对话框还提供了有关更改变量后将执行哪些配置脚本，以及作为更改的结果将启动哪些新服务等信息。YaST 将提示您确认更改，并通知您在选择完成退出对话框后将执行哪些脚本。在这里还可以选择需要现在跳过而在以后启动的服务和脚本。YaST 将自动应用所有的更改并重新启动涉及的所有服务以使更改生效。

8.3.2 手动更改系统配置

要手动更改系统配置，请执行如下操作

- 1 成为 `root` 用户。
- 2 使用 `telinit 1` 将系统转入单用户模式（运行级别 1）。
- 3 使用您选择的编辑器根据需要对配置文件进行更改。

如果不使用 YaST 来更改 `/etc/sysconfig` 中的配置文件，则要确保将空变量值用两个引号表示 (`KEYTABLE=""`)，并将含有空白的值用引号括起来。只包括一个单词的值不需要用引号括起来。

- 4 执行 `SUSEconfig` 来确保更改生效。
- 5 使用类似 `telinit default_runlevel` 的命令将系统返回到先前的运行级别。使用系统的默认运行级别替代 `default_runlevel`。如果想返回有网络和 X 的完全多用户方式，请选择 5；如果希望在有网络的完全多用户方式下工作，请选择 3。

这一过程主要用于更改整个系统范围的配置，例如网络配置。若要进行较小的更改，不一定要切换到单用户方式，但这样做可以完全确保正确重新启动所有相关的程序。

提示: 配置自动系统配置

要禁用 `SUSEconfig` 设定的自动系统配置，请将 `/etc/sysconfig/suseconfig` 中的变量 `ENABLE_SUSECONFIG` 设置为 `no`。如果要使用 `SUSE` 安装支持，请不要禁用 `SUSEconfig`。也可以部分禁用自动配置。

引导加载程序 GRUB

本章描述如何配置 GRUB（SUSE® Linux Enterprise Desktop 中使用的引导加载程序）。一个特殊的 YaST 模块可用于执行所有设置。如果您不熟悉在 Linux 中进行引导的相关内容，请阅读下面几节获得一些背景信息。本章还介绍了使用 GRUB 进行引导时经常遇到的一些问题和它们的解决方案。

注意: 使用 UEFI 的计算机上无 GRUB

配有传统 BIOS 的计算机和使用兼容支持模块 (CSM) 的 UEFI（统一可扩展固件接口）计算机上都例行安装了 GRUB。在未启用 CSM 的 UEFI 计算机上，将自动安装 eLILO（假设 DVD1 已成功引导）。请参见系统上 `/usr/share/doc/packages/elilo/` 处的 eLILO 文档以了解细节。

本章主要介绍引导加载程序 GRUB 的引导管理和配置。[第8章 引导和配置 Linux 系统](#) [63] 中将引导过程作为一个整体进行了介绍。引导加载程序代表计算机 (BIOS) 和操作系统 (SUSE Linux Enterprise Desktop) 之间的接口。引导加载程序的配置直接影响到操作系统的启动。

本章经常出现以下术语，可能需要进行解释：

主引导记录

MBR 的结构是由独立于操作系统的约定定义的。前 446 个字节为程序代码保留。它们通常保存部分引导加载程序或操作系统选择器。随后的 64 个字节为最多包含 4 项的分区表提供空间。分区表包含有关硬盘分区和文件系统类型的信息。操作系统需要使用此表来处理硬盘。如果 MBR 中有传统通用代码，则只应将一个分区标记为活动。MBR 的最后两个字节包含静态“幻

数”(AA55)。一些 BIOS 会将包含不同值的 MBR 视为无效，因此引导时不会考虑此 MBR。

引导扇区

引导扇区是硬盘分区（除扩展分区之外）上的前几个扇区，扩展分区只充当其他分区的“容器”。引导扇区具有 512 字节的空间，引导扇区储存用于引导安装在各个分区上的操作系统的代码。这适用于经过格式化的 DOS、Windows 和 OS/2 分区的引导扇区，这些扇区还包含文件系统的一些重要的基本数据。相比之下，Linux 分区的引导扇区在设置文件系统（而不是 XFS）之后最初是空的。因此，即使 Linux 分区包含内核和有效的 root 文件系统，它也不能通过自身进行引导。储存了引导系统的有效代码的引导扇区具有与 MBR 中的最后两个字节 (AA55) 相同的幻数。

9.1 通过 GRUB 引导

GRUB (Grand Unified Bootloader) 由两段组成。Stage 1 包含 512 个字节，它的唯一任务就是装载引导加载程序的第二段。随后，装载 stage 2。这一段包含引导加载程序的主要部分。

在一些配置中，可以使用中间段 1.5，它能从适当的文件系统中找到并装载第二段。如果可能，将在安装时或使用 YaST 初始设置 GRUB 时默认选择此方法。

Stage 2 可以访问多个文件系统。当前，支持 Ext2、Ext3、ReiserFS、Minix，以及 Windows 使用的 DOS FAT 文件系统。在某种程度上还支持 BSD 系统使用的、XFS、UFS 和 FFS。从版本 0.95 开始，GRUB 还能够从包含 ISO 9660 标准文件系统、符合“El Torito”规范的 CD 或 DVD 进行引导。即使是在引导系统之前，GRUB 也可以访问支持的 BIOS 磁盘设备（BIOS 检测到的软盘或硬盘、CD 驱动器和 DVD 驱动器）的文件系统。因此，对 GRUB 配置文件 (menu.lst) 进行更改不要求新安装引导管理器。当引导系统时，GRUB 重装载菜单文件以及内核或初始 ram 磁盘 (initrd) 的有效路径和分区数据，并对这些文件进行定位。

GRUB 的实际配置是基于三个文件进行的，下面对这三个文件进行介绍：

/boot/grub/menu.lst

此文件包含有关可通过 GRUB 进行引导的分区或操作系统的所有信息。没有这些信息，GRUB 命令行将提示用户如何继续（请参见“[在引导过程中编辑菜单项](#)”一节 [83]获取详细信息）。

```
/boot/grub/device.map
```

此文件将 GRUB 和 BIOS 符号中的设备名转换为 Linux 设备名。

```
/etc/grub.conf
```

此文件包含 GRUB shell 正确安装引导加载程序所需的命令、参数和选项。

可以通过多种方式控制 GRUB。可以在图形菜单（启动屏幕）中选择现有配置的引导项。配置是从文件 `menu.lst` 装载的。

在 GRUB 中，在引导前可以更改所有引导参数。例如，可以通过这种方式更正编辑菜单文件时出现的错误。还可以在输入提示符处以交互的方式输入引导命令（请参见“[在引导过程中编辑菜单项](#)”一节 [83]）。GRUB 能够在引导前确定内核和 `initrd` 的位置。通过这种方式，您甚至可以引导在引导加载程序配置中不存在任何项的已安装操作系统。

GRUB 实际上以两个版本存在：作为引导加载程序以及作为 `/usr/sbin/grub` 中的普通 Linux 程序。此程序被称为 *GRUB shell*。它在已安装系统中提供 GRUB 的仿真，并且可用来安装 GRUB 或在应用新设置之前对其进行测试。将 GRUB 作为引导加载程序安装在硬盘或软盘上的功能以 `install` 和 `setup` 命令的形式集成在 GRUB 中。当装载了 Linux 后在 GRUB shell 中可用。

9.1.1 GRUB 引导菜单

带有引导菜单的图形启动屏幕基于 GRUB 配置文件 `/boot/grub/menu.lst`，该文件包含有关可以通过菜单引导的所有分区或操作系统的所有信息。

每次引导系统时，GRUB 都从文件系统装载菜单文件。出于此原因，不必每次更改文件后都重安装 GRUB。使用 YaST 引导加载程序修改 GRUB 配置，如第 9.2 节“[使用 YaST 配置引导加载程序](#)”[85] 中所述。

菜单文件中包含命令。语法非常简单。每行都包含一条命令，后跟可选参数，可选参数之间用空格隔开，就像在 shell 中一样。出于历史原因，某些命令允许在第一个参数前使用 `=`。注释以井号 (`#`) 开头。

若要在菜单概述中标识菜单项，请为每项设置一个 `title`。关键字 `title` 后的文本（包括任何空格）显示为菜单中的可选择选项。当选择此菜单项时，将执行下一个 `title` 前的所有命令。

最简单的情况是重定向到其他操作系统的引导加载程序。命令是 `chainloader`，参数通常是 GRUB 中另一个分区的引导块 `block notation`。例如：

```
chainloader (hd0,3)+1
```

GRUB 中的设备名在“[硬盘和分区的命名约定](#)”一节 [80] 中有所解释。此示例指定第一个硬盘第四个分区中的第一个块。

使用命令 `kernel` 指定内核映像。第一个参数是指向分区中内核映像的路径。命令行上的其他参数将被传递到内核。

如果内核不具有访问根分区的内置驱动程序，或者使用了具有高级热插拨功能的最新 `linux` 系统，则必须用单独的 GRUB 命令指定 `initrd`，该命令的唯一参数便是指向 `initrd` 文件的路径。因为 `initrd` 的装载地址会被写入装载的内核映像中，所以 `initrd` 命令必须紧接在 `kernel` 命令之后。

命令 `root` 简化了内核和 `initrd` 文件的指定。`root` 的唯一参数是一个设备或分区。此设备用于所有内核、`initrd` 或下一个 `root` 命令前未显式指定设备的其他文件路径。

每个菜单项的末尾都间接指定 `boot` 命令，因此无需将其写入菜单文件中。但是，如果以交互方式使用 GRUB 进行引导，则必须在最后输入 `boot` 命令。该命令本身没有参数。它只引导装载的内核映像或指定的链装载程序。

在写入所有菜单项之后，将其中一项定义为 `default` 项。否则，将使用第一项（项 0）。您还可以指定在一段时间后引导默认项的超时值（以秒为单位）。`timeout` 和 `default` 通常在各菜单项前面。示例文件在“[示例菜单文件](#)”一节 [81] 中有所介绍。

硬盘和分区的命名约定

GRUB 用于硬盘和分区的命名约定不同于普通 Linux 设备使用的命名约定。它更类似于 BIOS 执行的简单磁盘枚举，而语法类似于一些 BSD 衍生程序中使用的语法。在 GRUB 中，分区的编号从 0 开始。它表示 `(hd0, 0)` 是第一块硬盘的第一个分区。在普通台式机上，作为 `Primary Master`（第一个 IDE 控制器上的主设备）连接的硬盘所对应的 Linux 设备名为 `/dev/sda1`。

4 个可能的主分区所分配的分区号为 } 0 到 3。逻辑分区的编号从 4 开始：

```
(hd0,0)    first primary partition of the first hard disk
(hd0,1)    second primary partition
(hd0,2)    third primary partition
(hd0,3)    fourth primary partition (usually an extended partition)
(hd0,4)    first logical partition
(hd0,5)    second logical partition
```

GRUB 依赖于 BIOS 设备，它不区分 IDE、SATA、SCSI 和硬件 RAID 设备。BIOS 或其他控制器识别的所有硬盘将按照 BIOS 中显示的引导顺序进行编号。

不过，通常不能将 Linux 设备名准确映射为 BIOS 设备名。它借助某种算法生成这一映射并将其保存到文件 `device.map` 中，可以根据需要对该文件进行编辑。有关文件 `device.map` 的信息在 [第 9.1.2 节“文件 device.map”](#) [83] 中有所介绍。

完整的 GRUB 路径包含写在括号中的设备名和指向指定分区的文件系统中文件的路径。路径以斜线开头。例如，在具有一个 IDE 硬盘（该硬盘的第一个分区中包含 Linux）的系统上，可以按如下方式指定可引导内核：

```
(hd0,0)/boot/vmlinuz
```

示例菜单文件

以下示例描述了 GRUB 菜单文件的结构。该示例安装包括 `/dev/sda5` 下的 Linux 引导分区、`/dev/sda7` 下的根分区和 `/dev/sda1` 下的 Windows 安装。

```
gfxmenu (hd0,4)/boot/message
color white/blue black/light-gray
default 0
timeout 8

title linux
    root (hd0,4)
    kernel /boot/vmlinuz root=/dev/sda7 vga=791 resume=/dev/sda9
    initrd /boot/initrd

title windows
    rootnoverify (hd0,0)
    chainloader +l

title floppy
    rootnoverify (hd0,0)
    chainloader (fd0)+l

title failsafe
    root (hd0,4)
    kernel /boot/vmlinuz.shipped root=/dev/sda7 ide=nodma \
```

```
apm=off acpi=off vga=normal nosmp maxcpus=0 3 noresume  
initrd /boot/initrd.shipped
```

第一块定义了启动屏幕的配置：

gfxmenu (hd0,4)/message

背景图像 `message` 位于 `/dev/sda5` 分区的顶级目录中。

color white/blue black/light-gray

色彩模式：白色（前景色）、蓝色（背景色）、黑色（所选内容）、浅灰色（所选内容的背景）。颜色方案对启动屏幕没有任何影响，它只影响通过按 **Esc** 键退出启动屏幕后所访问的可自定义的 **GRUB** 菜单。

default 0

第一个菜单项 `title linux` 是默认情况下引导的对象。

timeout 8

如果 8 秒钟后无任何用户输入，**GRUB** 将自动引导默认项。要检测自动引导，请删除 `timeout` 行。如果设置 `timeout 0`，**GRUB** 将立即引导默认项。

第二块（也就是最大的块）列出了各个可引导的操作系统。各个操作系统的不同部分由 `title` 引出。

- 第一项 (`title linux`) 负责引导 **SUSE Linux Enterprise Desktop**。内核 (`vmlinux`) 位于第一块硬盘的第一个逻辑分区（引导分区）。内核参数（例如引导分区和 **VGA** 方式）也被追加在此处。根分区是根据 **Linux** 命名约定 (`/dev/sda7`) 指定的，因为此信息将由内核读取而与 **GRUB** 无关。`initrd` 也位于第一块硬盘的第一个逻辑分区中。
- 第二项负责装载 **Windows**。**Windows** 将从第一块硬盘的第一个分区 (`hd0, 0`) 引导。命令 `chainloader +1` 将导致 **GRUB** 读取并执行指定分区的第一个扇区。
- 下一项支持从软盘进行引导，而无需修改 **BIOS** 设置。
- 引导选项 `failsafe` 用一组内核参数启动 **Linux**，这些参数使 **Linux** 甚至可以在有问题的系统上引导。

随时可以根据需要更改菜单文件。**GRUB** 会在下次引导时使用修改后的设置。使用 **YaST** 或所选的编辑器对文件进行永久编辑。或者，使用 **GRUB** 的编辑功

能可以按交互方式进行临时更改（请参见“在引导过程中编辑菜单项”一节 [83]）。

在引导过程中编辑菜单项

在图形引导菜单中，使用箭头键选择要引导的操作系统。如果选择 Linux 系统，则可以在引导提示符处输入其他引导参数。若要直接编辑个别菜单项，请按 **Esc** 键退出启动屏幕并进入 GRUB 基于文本的菜单，然后按 **E** 键。通过这种方式进行的更改仅适用于当前引导，不会被永久采用。

重要：引导过程中的键盘布局

US 键盘布局是引导时唯一可用的键盘布局。请参见图 6.1 “US Keyboard Layout”（↑系统分析和微调指南）的图。

编辑菜单条目简化了无法再进行引导的有问题系统的修复工作，因为可以通过手动输入参数规避引导加载程序中有关问题的配置文件。在引导过程中手动输入参数还可用于测试新设置而避免损坏本机系统。

在激活编辑方式后，可以使用箭头键选择要编辑其配置的菜单项。若要使配置可以编辑，请再次按 **E** 键。通过这种方式，可以编辑不正确的分区或路径指定，从而防止它们对引导进程产生负面影响。按 **Enter** 键退出编辑方式并返回菜单。随后按 **B** 键引导此项。可以进行的进一步操作显示在底部的帮助文本中。

若要永久输入更改的引导选项并将它们传递到内核，则以 `root` 用户身份打开文件 `menu.lst` 并将相应的内核参数追加到现有的行上，用空格分隔：

```
title linux
    root(hd0,0)
    kernel /vmlinuz root=/dev/sda3 additional parameter
    initrd /initrd
```

GRUB 会在下次引导系统时自动采用新参数。或者，还可以通过 YaST 引导加载程序模块进行此更改。将新参数追加到现有的行上，用空格分隔。

9.1.2 文件 device.map

文件 `device.map` 将 GRUB 和 BIOS 设备名映射为 Linux 设备名。在包含 IDE 和 SCSI 硬盘的混合系统中，GRUB 必须通过特殊过程尝试确定引导顺序，因为 GRUB 不能访问 BIOS 上有关引导顺序的信息。GRUB 会将此分析的结果保存

在文件 `/boot/grub/device.map` 中。对于 BIOS 中引导顺序设置为 IDE 在 SCSI 之前的系统，文件 `device.map` 如下所示：

```
(fd0)  /dev/fd0
(hd0)  /dev/sda
(hd1)  /dev/sdb
```

因为 IDE、SCSI 和其他硬盘的顺序取决于不同的因素，并且 Linux 无法标识映射，所以可以在 `device.map` 文件中手动设置顺序。如果在引导时遇到问题，则检查此文件中的顺序是否对应于 BIOS 中的顺序，如果需要，使用 GRUB 提示符 对其进行临时修改。引导了 Linux 系统之后，便可以使用 YaST 引导加载程序模块或所选的编辑器对文件 `device.map` 进行永久编辑。

在手动更改 `device.map` 之后，请执行以下命令重安装 GRUB。此命令导致重装文件 `device.map` 并且执行 `grub.conf` 中列出的命令：

```
grub --batch < /etc/grub.conf
```

9.1.3 文件 `/etc/grub.conf`

除了 `menu.lst` 和 `device.map` 之外，第三个重要的 GRUB 配置文件就是 `/etc/grub.conf`。此文件包含 GRUB shell 正确安装引导加载程序所需的命令、参数和选项：

```
setup --stage2=/boot/grub/stage2 --force-lba (hd0,1) (hd0,1)
quit
```

该命令将告知 GRUB 使用位于第一个硬盘 (hd0,1) 的第二个分区上的引导映像将引导加载程序自动安装到同一分区上。需要 `--stage2=/boot/grub/stage2` 参数在已装入的文件系统中安装 stage2 映像。一些 BIOS 具有不正确的 LBA 支持实施，`--force-lba` 提供了解决方案以忽略这些故障。

9.1.4 设置引导密码

即使是在引导操作系统之前，GRUB 也支持对文件系统的访问。没有根权限的用户可以访问 Linux 系统中的文件，而一旦引导系统后，他们将无权访问这些文件。若要阻止这种访问或防止用户引导某些操作系统，可以设置引导密码。

重要: 引导密码和启动屏幕

如果对 GRUB 使用引导密码，则不显示通常的启动屏幕。

以 root 用户身份按如下步骤设置引导密码：

- 1 在根提示符处，使用 grub-md5-crypt 加密密码：

```
# grub-md5-crypt
Password: ****
Retype password: ****
Encrypted: $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

- 2 将经过加密的字符串粘贴到 menu.lst 文件的全局部分：

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
password --md5 $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

现在，只有在按 P 键并输入密码后，才可以在引导提示符处执行 GRUB 命令。但是，用户仍可以从引导菜单引导所有操作系统。

- 3 要防止从引导菜单引导一个或多个操作系统，请将项 lock 添加到 menu .lst 中不输入密码就不能引导的每个部分。例如：

```
title linux
    kernel (hd0,4)/vmlinuz root=/dev/sda7 vga=791
    initrd (hd0,4)/initrd
    lock
```

在重引导系统并从引导菜单中选择 Linux 项后，将显示以下错误消息：

```
Error 32: Must be authenticated
```

按 Enter 键进入该菜单。然后按 P 键，系统将提示您输入密码。在输入密码并按 Enter 键之后，将引导所选的操作系统（在本例中为 Linux）。

9.2 使用 YaST 配置引导加载程序

在您的 SUSE Linux Enterprise Desktop 系统中配置引导加载程序的最简单方式就是使用 YaST 模块。在 YaST 控制中心，选择系统 > 引导加载程序。如图 9.1

“Boot Loader 设置” [86]中所示，它将显示您系统的当前引导加载程序配置，并允许您进行更改。

图 9.1 *Boot Loader 设置*



使用**扇区管理**选项卡可编辑、更改和删除单个操作系统的引导加载程序扇区。若要添加某个选项，请单击**添加**。要更改现有选项的值，请用鼠标选中它，然后单击**编辑**。要删除现有的条目，请选择它，单击**删除**。如果对引导加载程序选项不熟悉，请先阅读**第 9.1 节 “通过 GRUB 引导”** [78]。

使用**引导加载程序安装**选项卡查看并更改类型、位置和高级装载程序设置的相关设置。

从单击**其他**后打开的下拉菜单中访问高级配置选项。内置编辑器让您可以更改 GRUB 配置文件（细节请参见**第 9.1 节 “通过 GRUB 引导”** [78]）。也可以删除现有配置并从头开始或让 YaST 建议新配置。也可以向磁盘写入配置或从磁盘重新读取配置。要恢复在安装期间保存的原始主引导记录 (MBR)，请选择**恢复硬盘的主引导记录**。

9.2.1 调整默认引导项

要更改默认引导的系统，请按如下所示继续：

过程 9.1 设置默认系统

- 1 打开扇区管理选项卡。
- 2 从列表中选择所需的条目。
- 3 单击 设为默认。
- 4 单击 完成以激活这些更改。

9.2.2 修改引导加载程序位置

要修改引导加载程序的位置，请遵循以下步骤：

过程 9.2 更改引导加载程序位置

- 1 选择引导加载程序安装选项卡，然后为引导加载程序位置选择以下某个选项：

从引导分区引导

/boot 分区的引导扇区。

从扩展分区引导

这将在扩展分区容器中安装引导加载程序。

从主引导记录引导

本操作会在第一个磁盘的 MBR 中安装引导加载程序（根据 BIOS 中预设的引导顺序）。

从引导分区引导

这将在 / 分区的引导扇区安装引导加载程序。

自定义引导分区

手动使用此选项来指定引导加载程序的位置。

- 2 单击 结束来应用更改。

9.2.3 更改引导加载程序超时值

引导加载程序不会立即引导默认系统。超时期间，可以选择要引导的系统或编写一些内核参数。要设置引导加载程序超时值，请执行如下操作：

过程 9.3 更改引导加载程序超时值

- 1 打开引导加载程序安装选项卡。
- 2 单击引导加载程序选项。
- 3 通过输入新值、用鼠标单击相应的箭头键或使用键盘上的箭头键来更改超时（以秒为单位）的值。
- 4 单击确定。
- 5 单击完成以保存更改。

9.2.4 设置引导密码

使用此 YaST 模块，还可以设置密码来保护引导。这提供了更高的安全性级别。

过程 9.4 设置引导加载程序密码

- 1 打开引导加载程序安装选项卡。
- 2 单击引导加载程序选项。
- 3 在菜单界面密码中设置密码。
- 4 单击确定。
- 5 单击完成以保存更改。

9.2.5 调整磁盘顺序

如果您的计算机有多个硬盘，则可以按照计算机的 BIOS 设置指定磁盘引导顺序（请参见第 9.1.2 节“文件 `device.map`” [83]）。为此，请执行如下操作：

过程 9.5 设置磁盘顺序

- 1 打开引导加载程序安装选项卡。
- 2 单击引导加载程序安装详细信息。
- 3 如果列出了多个磁盘，请选择一个，然后单击*向上*或*向下*来对显示的磁盘重新排序。
- 4 单击*确定*保存更改。
- 5 单击*完成*以保存更改。

9.2.6 配置高级选项

高级引导选项可以通过*引导加载程序安装 > 引导加载程序选项*来配置。通常情况下，不需要更改默认设置。

在分区表中设置活动标志，以引导分区

激活包含该引导加载程序的分区。一些旧操作系统（如 Windows 98）只能从活动分区中引导。

调试标志

在调试模式中设置 GRUB，该模式可显示表明磁盘活动的消息。

将通用引导代码写入 MBR 中

使用通用、独立于操作系统的代码替换当前 MBR。

隐藏引导菜单

隐藏引导菜单并引导默认项。

使用受信任的 GRUB

启动支持受信任计算功能的受信任 GRUB。

图形菜单文件

显示引导屏幕时使用的图形文件的路径。

串行连接参数

如果您的计算机是通过串行控制台控制的，则可指定要使用的 COM 端口及其运行速度。还可以将终端定义设置为“串行”。有关细节，请参见 `info`

grub 或 <http://www.gnu.org/software/grub/manual/grub.html>。

终端定义

如果您正在通过串行控制台引导，则在此输入“串行”，否则请保留为空。在这种情况下，您还需要指定串行连接参数。

9.2.7 更改引导加载程序类型

在引导加载程序安装中设置引导加载程序类型。SUSE Linux Enterprise Desktop 中的默认引导加载程序是 GRUB。如要使用 LILO，请执行如下操作：

过程 9.6 更改引导加载程序类型

- 1 选择引导加载程序安装选项卡。
- 2 对于引导加载程序，请选择 *LILO*。
- 3 在打开的对话框中，选择以下某个操作：

建议新配置

让 YaST 推荐一个新的配置。

转换当前配置

让 YaST 转换当前的配置。在转换配置时，有些设置可能会丢失。

从头开始新的配置

编写自定义配置。此操作在安装 SUSE Linux Enterprise Desktop 期间不可用。

读取保存在磁盘上的配置

装载自己的 `/etc/lilo.conf`。此操作在安装 SUSE Linux Enterprise Desktop 期间不可用。

- 4 单击确定保存更改
- 5 在主对话框中单击完成以应用更改。

转换时，旧的 GRUB 配置将保存到磁盘上。如要使用它，只需将引导加载程序类型改回 GRUB，然后选择恢复转换前保存的配置。此操作仅在已安装的系统上可用。

注意: 自定义引导加载程序

如果想要使用 GRUB 或 LILO 以外的引导加载程序，请选择不安装任何引导加载程序。在选择该选项之前，请仔细阅读您的引导加载程序文档。

9.3 卸载 Linux 引导加载程序

YaST 可用于卸载 Linux 引导加载程序并将 MBR 恢复为安装 Linux 之前的状态。在安装过程中，YaST 自动创建原始 MBR 的备份副本并根据请求进行恢复。

要卸载 GRUB，请启动 YaST 引导加载程序模块（系统 > 引导加载程序）。选择其他 > 恢复硬盘的主引导记录然后选择是，重写加以确认。

9.4 创建引导 CD

如果使用引导管理器引导系统时出现问题或如果不能将引导管理器安装在硬盘的 MBR 或软盘上，那么还可以创建包含所有必需的 Linux 启动文件的可引导 CD。这需要您的系统中安装有 CD 刻录机。

用 GRUB 创建可引导 CD-ROM 只需要特殊形式的 *stage2*（名为 *stage2_eltorito*）以及自定义的 *menu.lst*（可选）。不需要标准文件 *stage1* 和 *stage2*。

过程 9.7 创建引导 CD

1 将目录更改为要创建 ISO 映像的目录，例如：`cd /tmp`

2 创建 GRUB 的子目录，并更改为新创建的 *iso* 目录：

```
mkdir -p iso/boot/grub && cd iso
```

3 将内核、文件 *stage2_eltorito*、*initrd*、*menu.lst* 和 *message* 复制到 *iso/boot/*：

```
cp /boot/vmlinuz boot/
cp /boot/initrd boot/
cp /boot/message boot/
cp /usr/lib/grub/stage2_eltorito boot/grub
cp /boot/grub/menu.lst boot/grub
```

- 4 调整 `boot/grub/menu.lst` 中的路径入口，使它们指向 **CD-ROM** 设备。执行此操作的方法是将路径名中硬盘的设备名（以 `(hdx,y)` 格式列出）替换为 **CD-ROM** 驱动器的设备名（`cd`）。您可能还需要调整消息文件、内核和 `initrd` 的路径，它们应分别指向 `/boot/message`、`/boot/vmlinuz` 和 `/boot/initrd`。调整后，`menu.lst` 的显示应与以下示例类似：

```
timeout 8
default 0
gfxmenu (cd)/boot/message

title Linux
    root (cd)
    kernel /boot/vmlinuz root=/dev/sda5 vga=794 resume=/dev/sda1 \
    splash=verbose showopts
    initrd /boot/initrd
```

使用 `splash=silent` 代替 `splash=verbose` 来防止引导过程中出现引导消息。

- 5 用以下命令创建 ISO 映像：

```
genisoimage -R -b boot/grub/stage2_eltorito -no-emul-boot \
-boot-load-size 4 -boot-info-table -iso-level 2 -input-charset utf-8 \
-o grub.iso /tmp/iso
```

- 6 使用您选择的实用程序将最终文件 `grub.iso` 烧录到 **CD** 上。不要将 ISO 映像作为数据文件烧录，而要使用烧录实用程序中烧录 **CD** 映像的选项。

9.5 图形 SUSE 屏幕

如果将选项 `vga=value` 用作内核参数，则会在第一个控制台上显示图形 **SUSE** 屏幕。如果您使用 **YaST** 进行安装，则将依照所选的分辨率和图形卡自动激活此选项。可以根据需要通过三种方法禁用 **SUSE** 屏幕：

在必要时禁用 SUSE 屏幕。

在命令行上输入命令 `echo 0 >/proc/splash` 以禁用图形屏幕。要将其再次激活，请输入 `echo 1 >/proc/splash`。

默认禁用 SUSE 屏幕。

将内核参数 `splash=0` 添加到您的引导加载程序配置中。[第 9 章 引导加载程序 GRUB](#) [77] 提供了有关此内容的详细信息。但是，如果您倾向于使用文本方式（这是早期版本中的默认方式），请设置 `vga=normal`。

完全禁用 SUSE 屏幕

编译新内核并禁用帧缓冲支持中的选项使用启动屏幕而不是引导徽标。

提示

在内核中禁用帧缓冲支持也会自动禁用启动屏幕。如果您使用自定义内核运行 SUSE，则它不能为系统提供任何支持。

9.6 查错

本节列出使用 GRUB 进行引导的一些常见问题并提供可能解决方案的简短描述。在位于支持数据库 <http://support.novell.com/> 的文章中介绍了其中一些问题。用搜索对话框搜索 *GRUB*、*引导*和*引导加载程序*之类的关键词。

GRUB 和 XFS

XFS 未在分区引导块中为 `stage1` 预留任何空间。因此，不要指定 XFS 分区作为引导加载程序的位置。此问题可以通过创建单独的引导分区（不使用 XFS 进行格式化）得到解决。

GRUB 报告 GRUB Geom 错误

当引导系统时，GRUB 将检查连接的硬盘的磁盘空间。有时，BIOS 将返回不一致的信息，GRUB 将报告 GRUB Geom 错误。如果出现这种情况，请更新 BIOS。

如果将 Linux 安装在未在 BIOS 中注册的其他硬盘上，GRUB 也会返回此错误消息。找到并正确装载了引导加载程序的 `stage1`，但未找到 `stage2`。可以通过在 BIOS 中注册新硬盘解决此问题。

包含多个硬盘的系统不会进行引导

安装时，YaST 可能没有正确确定硬盘的引导顺序。例如，GRUB 可能将 IDE 磁盘视为 `hd0`，将 SCSI 磁盘视为 `hd1`，尽管 BIOS 中的引导顺序是相反的（SCSI 先于 IDE）。

在这种情况下，在引导进程中借助 GRUB 命令行对硬盘进行更正。在引导系统后，编辑 `device.map` 永久应用新映射。然后，检查 `/boot/grub/menu.lst` 和 `/boot/grub/device.map` 文件中的 GRUB 设备名，并使用以下命令重安装引导加载程序：

```
grub --batch < /etc/grub.conf
```

从第二块硬盘引导 Windows

某些操作系统（例如 Windows）只能从第一块硬盘进行引导。如果这样的操作系统安装在第一块硬盘之外的硬盘上，您可以影响相应菜单项的逻辑更改。

```
...
title windows
    map (hd0) (hd1)
    map (hd1) (hd0)
    chainloader (hd1,0)+1
...
```

在此示例中，将从第二块硬盘启动 Windows。出于此目的，请使用 `map` 更改硬盘的逻辑顺序。此更改不会影响 GRUB 菜单文件中的逻辑。因此，必须为 `chainloader` 指定第二块硬盘。

9.7 有关详细信息

有关 GRUB 的大量信息可以在 <http://www.gnu.org/software/grub/> 处获得。还请参见 grub 信息页面。您也可以在位于 <http://www.novell.com/support> 的支持数据库中搜索关键字“GRUB”获得有关特殊问题的信息。

特别的系统功能组件

本章提供有关各种软件包、虚拟控制台和键盘布局的信息。讨论诸如 `bash`、`cron` 和 `logrotate` 等软件组件，因为在最后的发行周期中已对这些组件进行了更改或增强。即使这些组件很小或者被认为不太重要，但是用户可能希望更改它们的默认行为，因为这些组件通常是与系统紧密结合的。本章的最后是有关语言和国家/地区特定设置（`I18N` 和 `L10N`）的内容。

10.1 特殊软件包的相关信息

程序 `bash`、`cron`、`logrotate`、`locate`、`ulimit` 和 `free` 对于系统管理员和许多用户是非常重要的。手册页和信息页是命令相关信息两个有用来源，但是它们并不是始终可用的。`GNU Emacs` 是一种流行的并且非常容易配置的文本编辑器。

10.1.1 `bash` 包和 `/etc/profile`

`Bash` 是默认的系统 shell。在用作登录 shell 时，它将读取几个初始化文件。`Bash` 按照这些文件在列表中出现的顺序处理它们：

1. `/etc/profile`
2. `~/.profile`
3. `/etc/bash.bashrc`
4. `~/.bashrc`

在 `~/.profile` 或 `~/.bashrc` 中进行自定义设置。要确保正确处理这些文件，需要将基本设置从 `/etc/skel/.profile` 或 `/etc/skel/.bashrc` 复制到用户的主目录中。建议在更新后从 `/etc/skel` 复制这些设置。执行以下 `shell` 命令可防止个人调整的损失：

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

然后从 `*.old` 文件将个人调整复制过来。

10.1.2 cron 包

如果要在预定义的时间在后台定期自动运行命令，请使用 `cron` 工具。`cron` 是由特殊格式的时间表驱动的。这些表有一部分是系统附带的，但如有需要，用户可以自行编写表。

`cron` 表位于 `/var/spool/cron/tabs` 中。`/etc/crontab` 用作系统范围的 `cron` 表。输入在时间表之后且在此命令之前运行此命令的用户名。在例 10.1 “`/etc/crontab` 中的项” [96] 中，输入的是 `root`。位于 `/etc/cron.d` 中的包特定的表具有相同的格式。请参见 `cron` 手册页 (`man cron`)。

例 10.1 `/etc/crontab` 中的项

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

不能通过调用命令 `crontab -e` 来编辑 `/etc/crontab`。必须直接将该文件装载到编辑器中，对其进行修改，然后保存。

许多包将 `shell` 脚本安装到目录 `/etc/cron.hourly`、`/etc/cron.daily`、`/etc/cron.weekly` 和 `/etc/cron.monthly` 中，它们的执行是由 `/usr/lib/cron/run-crons` 控制的。`/usr/lib/cron/run-crons` 每

隔 15 分钟在主表 (/etc/crontab) 中运行一次。这样可以确保在适当的时间运行可能被忽略的进程。

要运行 hourly、daily 或在自定义时间运行其他周期性维护脚本，请删除通常使用 /etc/crontab 项的时戳文件（请参见例 10.2 “/etc/crontab：删除时戳文件”[97]，它删除了每个整点之前的 hourly 和每天早上 2:14 的 daily 等）。

例 10.2 /etc/crontab：删除时戳文件

```
59 * * * * root rm -f /var/spool/cron/lastrun/cron.hourly
14 2 * * * root rm -f /var/spool/cron/lastrun/cron.daily
29 2 * * 6 root rm -f /var/spool/cron/lastrun/cron.weekly
44 2 1 * * root rm -f /var/spool/cron/lastrun/cron.monthly
```

或者，在 /etc/sysconfig/cron 中将 DAILY_TIME 设置为 cron.daily 应该启动的时间。MAX_NOT_RUN 的设置确保日常作业被触发运行，即使用户在很长时间内没有在指定的 DAILY_TIME 打开计算机。MAX_NOT_RUN 的最大值为 14 天。

为了清楚起见，将日常系统维护作业分布在多个脚本中。这些脚本包含在包 aaa_base 中。例如，/etc/cron.daily 中包含组件 suse.de-backup-rpmd、suse.de-clean-tmp 或 suse.de-cron-local。

10.1.3 日志文件：包 logrotate

有许多系统服务（守护程序）以及内核本身定期将系统状态和特定事件记录到日志文件中。这样，管理员可以定期检查系统在某一时刻的状态，识别错误或故障功能，并精确诊断它们。这些日志文件通常储存在 FHS 指定的 /var/log 中，文件大小每天都会增长。logrotate 包可以帮助控制这些文件的生长。

用文件 /etc/logrotate.conf 配置 logrotate。特别地，include 规范主要配置了其他要读取的文件。在 /etc/logrotate.d 中产生日志文件、安装的各个配置文件的程序。例如，随包 apache2(/etc/logrotate.d/apache2) 和 syslogd(/etc/logrotate.d/syslog) 一起提供的文件。

例 10.3 */etc/logrotate.conf* 的示例

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or wtmp - we'll rotate them here
#/var/log/wtmp {
#    monthly
#    create 0664 root utmp
#    rotate 1
#}

# system-specific logs may be also be configured here.
```

通过 **cron** 控制 **logrotate**，并通过 `/etc/cron.daily/logrotate` 每天对其进行调用。

重要

使用 `create` 选项可以读取管理员在 `/etc/permissions*` 中进行的所有设置。确保没有因个人修改而引起的冲突。

10.1.4 **locate** 命令

locate 是一个用于查找文件的命令，它不包括在已安装软件的标准范围内。如果需要，请安装包 `findutils-locate`。`updatedb` 进程将在每天晚上或引导系统约 15 分钟后自动启动。

10.1.5 ulimit 命令

使用 `ulimit (user limits)` 命令，可以对系统资源的使用设置限制并将这些信息显示出来。`ulimit` 尤其适用于限制应用程序可用的内存。使用此命令，可以防止某个应用程序自己占用太多内存，这可能导致系统停顿。

可以对 `ulimit` 使用多个选项。要限制使用内存，请使用 [表 10.1 “ulimit：为用户设置资源”](#) [99] 中列出的选项。

表 10.1 *ulimit：为用户设置资源*

-m	最大驻留集大小
-v	壳层可用虚拟内存的最大量
-s	堆栈的最大大小
-c	创建的核心文件的最大大小
-a	所有当前限制均已报告

可以在 `/etc/profile` 中创建系统范围的项。在这里可以创建编程人员进行调试所需的核心文件。普通用户不能增加系统管理员在 `/etc/profile` 中指定的值，但可以在 `~/.bashrc` 中进行特殊输入。

例 10.4 *ulimit：~/.bashrc 中的设置*

```
# Limits maximum resident set size (physical memory):
ulimit -m 98304

# Limits of virtual memory:
ulimit -v 98304
```

必须以 KB 为单位指定内存大小。有关详细信息，请参见 `man bash`。

重要

并非所有 `shell` 都支持 `ulimit` 指令。如果您依赖于这些限制的内含设置，则 `PAM`（例如 `pam_limits`）提供了全面的调整功能。

10.1.6 free 命令

如果您的目的是查看当前使用了多少 RAM，则 `free` 命令可能会令人产生误解。`/proc/meminfo` 中提供了此信息。目前，使用 Linux 等现代操作系统的用户实际上无需过多地担心内存。可用 RAM 的概念可追溯到统一内存管理之前。可用内存不是好的内存这种说法非常适用于 Linux。因此，Linux 一直在平衡缓存方面下功夫，不允许实际上存在可用或未使用的内存。

内核基本上不直接管理任何应用程序或用户数据。而是在一个页缓存中管理应用程序和用户数据。如果内存不足，它的某些部分会被写入交换分区或文件中，借助于 `mmap` 命令，可以最先从这些交换分区或文件中读取这些部分（请参见 `man mmap`）。

此外，内核中还包含其他缓存，如 `slab` 缓存，其中储存着用于网络访问的缓存。这也许能够解释 `/proc/meminfo` 中计数器之间的差异。通过 `/proc/slabinfo` 可以访问大多数（但并非全部）`slab` 缓存。

10.1.7 手册页和信息页

对于某些 GNU 应用程序（如 `tar`），已不再保留手册页。对于这些命令，可使用 `--help` 选项快速查看信息页，其中提供更多深入的描述。`info` 是 GNU 的超文本系统。通过输入 `info info` 可以看到此系统的介绍。通过输入 `emacs -f Info` 可使用 Emacs 查看信息页，也可以在控制台中使用 `info` 直接查看信息页。还可以使用 `tkinfo`、`xinfo` 或帮助系统来查看信息页。

10.1.8 使用 man 命令选择手册页

通过 `man man-page`，通常可以显示便于阅读的手册页。现在，如果在不同部分中存在同名手册页，`man` 会提示用户，应显示哪个部分的页面，要求用户输入部分作为回答。

如果要返回到上一个行为，请在外壳初始化文件（如 `~/.bashrc`）中设置 `MAN_POSIXLY_CORRECT=1`。

10.1.9 GNU Emacs 的设置

GNU Emacs 是一个复杂的工作环境。下面几节介绍当启动 GNU Emacs 时处理的配置文件。有关详细信息，请参见 <http://www.gnu.org/software/emacs/>。

启动时，Emacs 读取包含用户、系统管理员和经销商的设置的多个文件以进行自定义或预配置。初始化文件 `~/.emacs` 被安装到 `/etc/skel` 中各个用户的主目录中。`.emacs` 又会读取文件 `/etc/skel/.gnu-emacs`。要自定义程序，请（通过 `cp /etc/skel/.gnu-emacs ~/.gnu-emacs`）将 `.gnu-emacs` 复制到用户主目录并在那里进行所需的设置。

`.gnu-emacs` 将文件 `~/.gnu-emacs-custom` 定义为 `custom-file`。如果用户通过 Emacs 中的 `customize` 选项进行设置，则这些设置将保存到 `~/.gnu-emacs-custom` 中。

通过 SUSE Linux Enterprise Desktop，emacs 包将文件 `site-start.el` 安装在目录 `/usr/share/emacs/site-lisp` 中。文件 `site-start.el` 在初始化文件 `~/.emacs` 之前进行装载。除其他作用之外，`site-start.el` 确保自动装载通过 Emacs 扩充包分发的特殊配置文件（例如 `psgml`）。此类型的配置文件也位于 `/usr/share/emacs/site-lisp` 中，总是以 `suse-start-` 开头。本地系统管理员可以在 `default.el` 中指定整个系统范围的设置。

初始化文件下的 EMACS 信息文件中提供了有关这些文件的详细信息：<info:/emacs/InitFile> 此位置还提供了有关如何禁止装载这些文件（如果需要）的信息。

Emacs 的部件被分成多个包：

- 基础包 `emacs`。
- `emacs-x11`（通常已安装）：支持 X11 的程序
- `emacs-nox`：不支持 X11 的程序。
- `emacs-info`：info 格式的联机文档。
- `emacs-el`：Emacs Lisp 中未编译的库文件。运行时不需要这些库文件。

- 如果需要，可安装众多附加包：emacs-auctex（用于 LaTeX）、psgml（用于 SGML 和 XML）、gnuserv（用于客户机和服务器操作）以及其他。

10.2 虚拟控制台

Linux 是一个多用户和多任务的系统。即使是在独立计算机系统上也可以感受到这些功能的好处。在文本方式下，提供了 6 个虚拟控制台。可以使用 Alt + F1 到 Alt + F6 在这些控制台间切换。第 7 个控制台是为 X 保留的，而第 10 个控制台显示内核消息。可以通过修改文件 /etc/inittab 指定更多的控制台或减少控制台。

要从 x 切换到控制台而不将其关闭，请使用 Ctrl + Alt + F1 到 Ctrl + Alt + F6。要返回到 X，请按 Alt + F7。

10.3 键盘映射

为了标准化程序的键盘映射，对以下文件进行了更改：

```
/etc/inputrc
/etc/X11/Xmodmap
/etc/skel/.Xmodmap
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/share/X11/app-defaults/XTerm
/usr/share/emacs/VERSION/site-lisp/term/*.el
```

这些更改只影响使用 terminfo 项的应用程序或其配置文件被直接更改（vi、less 等）的应用程序。不是系统附带的应用程序应该根据这些默认设置进行调整。

在 X 下，可以使用 Ctrl + Shift（右边的）访问组合键 (multikey)。同时可以在 /etc/X11/Xmodmap 中看到对应的项。

可以通过“X 键盘扩展”(XKB) 进行进一步的设置。桌面环境 GNOME (gswitchit) 和 KDE (kxkb) 也使用此扩展。

提示: 更多信息

有关 XKB 的信息, 请参见 `/etc/X11/xkb/README` 和那里列出的文档。

有关中文、日文和韩文 (CJK) 输入的详细信息, 请参见 Mike Fabian 的网页:
<http://www.suse.de/~mfabian/suse-cjk/input.html>

10.4 语言和国家/地区特定的设置

该系统在很大程度上实施了国际化, 可通过灵活的方式进行修改以满足本地需要。换句话说, 国际化 (*I18N*) 允许特定的本地化 (*L10N*)。I18N 和 L10N 这两个缩写词使用原单词的第一个和最后一个字母, 中间的数字表示省略的字母数。

设置是通过文件 `/etc/sysconfig/language` 中定义的 `LC_` 变量进行的。这不仅指本地语言支持, 还指消息 (语言)、字符集、排序顺序、日期和时间、数字和货币等类别。这些类别中的每一种都可以使用其自己的变量直接定义或使用文件 `language` 中的主变量间接定义 (请参见手册页 `man locale`)。

`RC_LC_MESSAGES`、`RC_LC_CTYPE`、`RC_LC_COLLATE`、`RC_LC_TIME`、
`RC_LC_NUMERIC`、`RC_LC_MONETARY`

这些变量以不带 `RC_` 前缀的形式传递到 `shell`, 它们代表所列出的类别。下面列出了相关 `shell` 配置文件。可以使用命令 `locale` 显示当前设置。

`RC_LC_ALL`

此变量 (如果设置) 将覆盖上述变量的值。

`RC_LANG`

如果未设置上述的任何变量, 则这是后备变量。默认情况下, 只设置 `RC_LANG`。这便于用户输入他们自己的值。

`ROOT_USES_LANG`

`yes` 或 `no` 变量。如果将其设置为 `no`, 则 `root` 用户始终在 `POSIX` 环境中工作。

这些变量可通过 YaST `sysconfig` 编辑器进行设置 (请参见第 8.3.1 节 “使用 YaST Sysconfig 编辑器更改系统配置” [75])。此类变量的值中包含语言代码、国家/地区代码、编码和修饰符。各部分之间通过特殊字符连接:

```
LANG=<language>[_<COUNTRY>].<Encoding>[@<Modifier>]]
```

10.4.1 一些示例

语言和国家/地区代码始终应该一起设置。语言设置遵循 ISO 639 标准（可从 <http://www.evertype.com/standards/iso639/iso639-en.html> 和 <http://www.loc.gov/standards/iso639-2/> 上获取）。国家/地区代码在 ISO 3166（可从 http://www.din.de/gremien/nas/nabd/iso3166ma/codlstpl/en_listpl.html 上获取）中列出。

只有设置可以在 `/usr/lib/locale` 中找到其可用描述文件的值才有意义。可以使用命令 `localedef` 基于 `/usr/share/i18n` 中的文件创建更多描述文件；描述文件是 `glibc-i18ndata` 包的一部分。可以使用以下命令创建 `en_US.UTF-8`（用于英国英语和美国英语）的描述文件：

```
localedef -i en_US -f UTF-8 en_US.UTF-8
```

```
LANG=en_US.UTF-8
```

如果在安装过程中选择的是美国英语，则这是默认设置。如果选择了其他语言，则将支持该语言，但仍使用 UTF-8 作为字符编码。

```
LANG=en_US.ISO-8859-1
```

这会将语言设置为英语，将国家/地区设置为美国，将字符集设置为 ISO-8859-1。此字符集不支持欧元符号，但它有时可用于尚未进行更新以支持 UTF-8 的程序。随后，Emacs 等程序将对定义字符集的字符串（在本例中为 ISO-8859-1）进行求值。

```
LANG=en_IE@euro
```

上例将欧元符号显式包含在语言设置中。严格来说，此设置目前已过时，因为 UTF-8 也包含欧元符号。该设置仅在应用程序不支持 UTF-8 但支持 ISO-8859-15 时才有用。

SUSEconfig 读取 `/etc/sysconfig/language` 中的变量并将必需的更改写入 `/etc/SuSEconfig/profile` 和 `/etc/SuSEconfig/csh.cshrc`。`/etc/SuSEconfig/profile` 被 `/etc/profile` 读取或用作其数据的来源。`/etc/SuSEconfig/csh.cshrc` 被用作 `/etc/csh.cshrc` 的数据来源。这使设置在整个系统范围内可用。

用户可以通过相应地编译他们的 `~/.bashrc` 覆盖系统默认值。例如，如果不想将整个系统范围的 `en_US` 用于程序消息，请包括 `LC_MESSAGES=es_ES`，这样消息将以西班牙语显示。

10.4.2 `~/.i18n` 中的语言环境设置

如果您对系统默认的区域设置不满意，请根据 **Bash** 脚本编写语法更改 `~/.i18n` 中的设置。`~/.i18n` 中的项覆盖来自 `/etc/sysconfig/language` 中的系统默认值。使用相同的变量名而不使用 `RC_` 名称空间前缀，例如，使用 `LANG` 而不是 `RC_LANG`：

```
LANG=cs_CZ.UTF-8
LC_COLLATE=C
```

10.4.3 语言支持的设置

消息类别中的文件通常只储存在对应的语言目录（例如 `en`）中以保留后备。如果将 `LANG` 设置为 `en_US` 并且 `/usr/share/locale/en_US/LC_MESSAGES` 中的消息文件不存在，则它将使用 `/usr/share/locale/en/LC_MESSAGES`。

还可以定义后备语言，例如，将布列塔尼语作为法语的后备语言，将加利西亚语作为葡萄牙语的后备语言。

```
LANGUAGE="br_FR:fr_FR"
```

```
LANGUAGE="gl_ES:es_ES:pt_PT"
```

如果需要，可改用挪威语变体 **Nynorsk** 和 **Bokmal**（将其他后备语言设置为 `no`）：

```
LANG="nn_NO"
```

```
LANGUAGE="nn_NO:nb_NO:no"
```

或

```
LANG="nb_NO"
```

```
LANGUAGE="nb_NO:nn_NO:no"
```

请注意，在挪威语中，LC_TIME 的处理方式也有所不同。

可能会出现一个问题，那就是无法正确识别用于分隔成组数位的分隔符。如果 LANG 设置为仅两个字母的语言代码（如 de），但使用的定义文件 glibc 位于 /usr/share/lib/de_DE/LC_NUMERIC，则将出现此问题。因此必须将 LC_NUMERIC 设置为 de_DE 以使系统能够识别出分隔符定义。

10.4.4 有关详细信息

- 《GNU C 库参考手册“》中的“区域设置和国际化”一章。”它包含在 glibc-info 中。
- Markus Kuhn 编写的 *Unix/Linux 的 UTF-8 和 Unicode 常见问题解答*，当前位于 <http://www.cl.cam.ac.uk/~mgk25/unicode.html>。
- *Unicode-Howto*，作者 Bruno Haible: /usr/share/doc/howto/en/html/Unicode-HOWTO.html。

打印机操作

SUSE® Linux Enterprise Desktop 支持用许多类型的打印机进行打印，包括远程网络打印机。打印机可以用 YaST 或手动进行配置。有关配置描述，请参见第 6.5 节 “Setting Up a Printer” (第 6 章 *Setting Up Hardware Components with YaST*, ↑部署指南)。启动和管理打印作业时既可以使用图形实用程序，也可以使用命令行实用程序。如果打印机未能按预期正常工作，请参见第 11.8 节 “查错” [116]。

CUPS 是 SUSE Linux Enterprise Desktop 中的标准打印系统。CUPS 高度面向用户。在很多情况下，它与 LPRng 兼容或者可以相对方便地进行调整。仅出于兼容性的原因，LPRng 包括在 SUSE Linux Enterprise Desktop 中。

可以根据接口（例如 USB 或网络）以及打印机语言对打印机进行区分。购买打印机时，请确认打印机具有一个您的硬件上可用的接口（比如 USB 或并行接口）和合适的打印机语言。可以按照以下三类打印机语言对打印机进行分类：

PostScript 打印机

Linux 和 Unix 中的内部打印系统使用 PostScript 这种打印机语言生成并处理大部分打印作业。这种语言已经有很长的历史并且非常有效。如果打印机可以直接处理 PostScript 文档而不需要在打印系统中通过附加步骤转换这些文档，则可以降低可能出现的错误的数目。因为 PostScript 打印机购买许可证要花费大量的成本，所以购买这些打印机的花费比不带 PostScript 解释器的打印机要高得多。

标准打印机（PCL 和 ESC/P 等语言）

虽然这些打印机语言有相当长的历史，但它们仍在进行扩展以处理打印机中的新功能。对于已知的打印机语言，打印系统可以借助 Ghostscript 将 PostScript 作业转换为相应的打印机语言。这一处理阶段被称为解释。最有名的语言有 PCL（主要是 HP 打印机及其克隆产品使用）和 ESC/P（Epson

打印机使用)。这些打印机语言通常受 Linux 支持，可以生成相当好的打印效果。Linux 可能不能处理非常新以及非常特别的打印机的一些功能，原因是开放源代码的开发人员可能仍在开发这些功能的代码。除了 HP 开发的 HPLIP 之外，当前尚没有其他打印机制造商开发 Linux 驱动程序并在开放源代码许可证下将这些驱动程序提供给 Linux 经销商。这些打印机中大多数价格适中。

专有打印机（也称作 GDI 打印机）

这些打印机不支持任何常见的打印机语言。这些打印机使用自己的无文档记录打印机语言，该语言在发布新版本时可能发生变化。通常只有 Windows 驱动程序供这些打印机使用。有关更多信息，请参见第 11.8.1 节“**打印机没有标准打印机语言支持**”[116]。

在您购买新打印机之前，请参考以下资源以了解您要购买的打印机的支持情况：

[http://www.linuxfoundation.org/en/OpenPrinting/
OpenPrinting.org](http://www.linuxfoundation.org/en/OpenPrinting/OpenPrinting.org) 打印机数据库。

[http://www.cs.wisc.edu/~ghost/
Ghostscript](http://www.cs.wisc.edu/~ghost/Ghostscript) Web 网页

`/usr/share/doc/packages/ghostscript-library/catalog.devices`
包含的驱动程序列表。

联机数据库总是显示最新的 Linux 支持状态。但是，Linux 分发只能集成生产时可用的驱动程序。因此，在最新的 SUSE Linux Enterprise Desktop 版本发布时，当前标为“完全支持”的打印机不一定具有此状态。这样，数据库不一定可以指出正确的状态，只是提供大致估计而已。

11.1 打印系统工作流程

用户创建一个打印作业。该打印作业包含有要打印的数据以及假脱机程序的信息，例如打印机的名称或打印机队列的名称，还可能包括过滤器的信息，例如打印机特定的选项。

每台打印机至少有一个专用打印机队列。假脱机程序储存着队列中的打印作业，直到所需打印机已做好接收数据的准备。打印机准备就绪后，假脱机程序通过过滤器和后端将数据发送到打印机。

过滤器将转换正在打印的应用程序生成的数据（通常为 PostScript 或 PDF，也可能为 ASCII、JPEG 等）特定于打印机的数据（PostScript、PCL、ESC/P 等）。PPD 文件中描述了打印机的功能。PPD 文件包含打印机特定的选项以及在打印机上启用这些选项所需的参数。过滤器系统用于确保用户选择的选项被启用。

如果使用的是 PostScript 打印机，则过滤器系统将数据转换为打印机特定的 PostScript。这样做不需要打印机驱动程序。如果使用的是非 PostScript 打印机，则过滤器系统将数据转换为打印机专用的数据。这样做需要一个适合您的打印机的打印机驱动程序。后端从过滤器接收打印机特定的数据，然后将其传递到打印机。

11.2 连接打印机的方法和协议

可以通过多种方法将打印机连接到系统。CUPS 打印系统的配置不能区分本地打印机和通过网络连接到系统的打印机。在 Linux 中，必须按照打印机制造商提供的手册中所描述的方法连接本地打印机。CUPS 支持串口、USB、并口和 SCSI 连接。

警告: 更改处于运行状态系统中的电缆连接

当将打印机连接到计算机时，一定不要忘记操作期间只能插入或拔下 USB 设备。为防止损坏系统或打印机，请在更改任何非 USB 连接前先关闭系统。

11.3 安装软件

PPD（PostScript 打印机描述）是描述属性（例如，分辨率）和选项（例如，双面打印单位的可用性）的计算机语言。这些描述对于使用 CUPS 中的各个打印机选项是必需的。如果没有 PPD 文件，打印数据将被以“原始”状态转发到打印机，通常这不是希望出现的情况。SUSE Linux Enterprise Desktop 安装期间将预安装多个 PPD 文件。

要配置 PostScript 打印机，最佳的方法是获得一个合适的 PPD 文件。包 `manufacturer-PPDs` 中提供许多 PPD 文件，标准安装会自动安装此包。请参见第 11.7.2 节“多种包中的 PPD 文件”[114]和第 11.8.2 节“没有合适的 PPD 文件可用于 PostScript 打印机”[117]。

可以将新 `ppd` 文件储存在目录 `/usr/share/cups/model/` 中或使用 YaST 添加到打印系统中（请参见“Adding Drivers with YaST”一节（第 6 章 *Setting Up Hardware Components with YaST*, ↑部署指南））。随后，可以在安装过程中选择 PPD 文件。

如果打印机制造商要求您除修改配置文件之外安装整个软件包，则一定要注意。首先，这种安装将导致丢失 SUSE Linux Enterprise Desktop 提供的支持；其次，打印命令将以不同的方式工作，系统可能不再能处理其他制造商的设备。出于此原因，不建议安装制造商软件。

11.4 网络打印机

网络打印机可以支持多种协议，其中某些甚至是同时进行的。虽然大多数支持的协议是标准化的，但某些制造商因为测试尚未正确实施标准的系统或要提供标准中未提供的功能，所以对标准进行了扩展（修改）。于是制造商提供仅用于几个操作系统的驱动程序，解决使用这些系统遇到的困难。不过很少提供 Linux 驱动程序。当前的情况是您在执行操作时不能假定每个协议都可以在 Linux 中正常工作。因此，您可能需要试验不同的选项来实现工作正常的配置。

CUPS 支持 `socket`、`LPD`、`IPP` 和 `smb` 协议。

套接字

套接字是指未先执行数据握手就将数据发送到因特网套接字所使用的连接。一些常用的套接字端口号包括 9100 或 35。设备 URI（统一资源标识符）的语法为 `socket://打印机 IP:端口`，例如 `socket://192.168.2.202:9100/`。

LPD（行式打印机守护程序）

RFC 1179 中对经过证明的 LPD 协议进行了介绍。在此协议下，在发送实际打印数据之前，将先发送一些与作业相关的数据，例如打印机队列的 ID。因此，在为数据传送配置 LPD 协议之前，必须指定打印机队列。不同打印机制造商的实施非常灵活，可以接受任何名称作为打印机队列。如果需要，打印机手册应该指出要使用的名称。通常使用 `LPT`、`LPT1`、`LP1` 或类似的名称。可以在不同 Linux 或 Unix 主机的 CUPS 系统中配置 LPD 队列。LPD 服务的端口号是 515。示例设备 URI 有 `lpd://192.168.2.202/LPT1`。

IPP（因特网打印协议）

IPP 是一个基于 HTTP 协议的相对较新的 (1999) 协议。使用 IPP，所传送的与作业有关的数据比其他协议要多一些。CUPS 使用 IPP 进行内部数据传送。这是在两个 CUPS 服务器之间转发队列的首选协议。要正确配置 IPP，必须提供打印队列的名称。IPP 的端口号是 631。示例设备 URI 有

`ipp://192.168.2.202/ps` 和

`ipp://192.168.2.202/printers/ps`。

SMB（Windows 共享）

CUPS 还支持在连接到 Windows 共享的打印机上进行打印。用于此目的的协议是 SMB。SMB 使用端口号 137、138 和 139。示例设备 URI 有

`smb://user:password@workgroup/smb.example.com/printer`、

`smb://user:password@smb.example.com/printer` 和

`smb://smb.example.com/printer`。

必须在配置之前确定打印机支持的协议。如果制造商未提供所需的信息，则可以使用命令 `nmap`（附带 `nmap` 包）来猜测协议。`nmap` 检查主机是否有打开的端口。例如：

```
nmap -p 35,137-139,515,631,9100-10000 printerIP
```

11.4.1 使用命令行工具配置 CUPS

除了用 YaST 设置 CUPS 选项，配置网络打印机时，CUPS 也可以用命令行工具进行配置，比如 `lpadmin` 和 `lpoptions`。您需要一个设备 URI，该 URI 由一个后端（例如并行端口）和多个参数组成。要确定系统上的有效设备 URI，请使用命令 `lpinfo -v | grep "://"`：

```
# lpinfo -v | grep "://"
direct usb://ACME/FunPrinter%20XL
direct parallel:/dev/lp0
```

使用 `lpadmin`，CUPS 服务器管理员可添加、删除或管理类和打印队列。要添加打印队列，请使用以下语法：

```
lpadmin -p queue -v device-URI -P PPD-file -E
```

使用指定的 PPD 文件 (`-p`)，则设备 (`-v`) 将用作队列 (`-P`)。这意味着如果要手动配置打印机，则必须了解 PPD 文件和设备 URI。

不要使用 `-E` 作为第一个选项。对于所有 CUPS 命令，将 `-E` 用作第一个参数设置使用加密连接。要启用打印机，必须使用 `-E`，如下面的示例所示：

```
lpadmin -p ps -v parallel:/dev/lp0 -P \
/usr/share/cups/model/Postscript.ppd.gz -E
```

以下示例配置了网络打印机：

```
lpadmin -p ps -v socket://192.168.2.202:9100/ -P \
/usr/share/cups/model/Postscript-levell.ppd.gz -E
```

有关 `lpadmin` 的更多选项，请参考 `lpadmin(1)` 的手册页。

在系统安装期间，某些选项被设置为默认值。可以为每个打印作业修改这些选项（根据所使用的打印工具）。也可以使用 **YaST** 来更改这些默认选项。使用命令行工具设置默认选项，如下所示：

1 首先，列出所有选项：

```
lpoptions -p queue -l
```

示例：

```
Resolution/Output Resolution: 150dpi *300dpi 600dpi
```

激活的默认选项通过加星号前缀 (*) 进行标识。

2 使用 `lpadmin` 更改选项：

```
lpadmin -p queue -o Resolution=600dpi
```

3 检查新设置：

```
lpoptions -p queue -l
```

```
Resolution/Output Resolution: 150dpi 300dpi *600dpi
```

普通用户运行 `lpoptions` 时，设置将写到 `~/.cups/lpoptions`。但是，根设置将写到 `/etc/cups/lpoptions`。

11.5 图形打印接口

像 `xpp` 这样的工具和 KDE 程序 `KPrinter` 都提供了一个图形界面，用于选择队列以及设置 `cups` 标准选项和通过 PPD 文件可用的打印机特定的选项。您甚至可

以使用 KPrinter 作为非 KDE 应用程序的标准打印接口。在这些应用程序的打印对话框中，指定 `kprinter` 或 `kprinter --stdin` 作为打印命令。使用的命令取决于应用程序的数据传输方式 — 只需尝试哪种方式起作用。如果正确设置，应用程序无论何时发布打印作业均应打开 KPrinter 对话框，这样您就可以使用此对话框来选择队列并设置其他打印选项了。这要求应用程序本身的打印设置不能与 KPrinter 的打印设置冲突，而且在启用 KPrinter 后，只能通过它来更改打印选项。有关 KPrinter 的详细信息，请参见第 6 章 *Managing Print Jobs* (↑*KDE 用户指南*)。

11.6 从命令行打印

要从命令行打印，请输入 `lp -d queuefilename`，使用相应的名称替换 `queuefilename` 和 `filename`。

有些应用程序依赖于 `lp` 命令来进行打印。在这种情况下，请在应用程序的打印对话框中输入正确的命令（通常无需指定 `filename`），例如 `lp -d queuefilename`。

11.7 SUSE Linux Enterprise Desktop 中的特殊功能

已对 CUPS 的许多功能进行了调整以用于 SUSE Linux Enterprise Desktop。这里将介绍一些最重要的更改。

11.7.1 CUPS 和防火墙

执行默认 SUSE Linux Enterprise Desktop 安装后，`Susefirewall2` 是活动的，且将网络接口配置为在外部区域中（该区域将阻止进来的流量）。使用 CUPS 时，必须调整这些默认设置。在第 9.4 节“`SuSEfirewall2`”（第 9 章 *Masquerading and Firewalls*, ↑*安全指南*）中提供了有关 `SUSEfirewall2` 配置的更多信息。

CUPS 客户程序

通常 CUPS 客户端在使用防火墙的可信网络环境中的常规工作站上运行。在这种情况下，建议将网络接口配置为在内部区域中，这样可以从网络内部访问工作站。

CUPS 服务器

如果 CUPS 服务器在受防火墙保护的可信网络环境中，则应将网络接口配置为在防火墙的内部区域中。建议不要在不可信网络环境中安装 CUPS 服务器，除非留心该服务器受到特殊防火墙规则和 CUPS 配置中的安全设置的保护。

11.7.2 多种包中的 PPD 文件

YaST 打印机配置仅使用系统上 `/usr/share/cups/model/` 中安装的 PPD 文件为 CUPS 设置队列。为查找用于某个打印机型号的合适的 ppd 文件，YaST 将在硬件检测过程中确定的供应商和型号与存在于系统上 `/usr/share/cups/model/` 中的所有 PPD 文件中的供应商和型号进行比较。为此，YaST 打印机配置根据从 PPD 文件抽取的供应商和型号信息生成一个数据库。选择打印机时，将收到与型号列表中的供应商和型号匹配的 PPD 文件。

仅使用 PPD 文件而不使用其他信息源的配置的优点在于可以随意修改 `/usr/share/cups/model/` 中的 PPD 文件。YaST 打印机配置可以识别更改并重生成为供应商和型号数据库。例如，如果您具有 `postscript` 打印机，通常您不需要 `cups-drivers` 包中的 `Foomatic` PPD 文件或 `gutenprint` 包中的 `Gutenprint` PPD 文件。而可以将您的 `PostScript` 打印机的 PPD 文件直接复制到 `/usr/share/cups/model`（如果它们尚不存在于 `manufacturer-ppds` 包中）以实现打印机的最佳配置。

cups 包中的 CUPS PPD 文件

为 `PostScript` 级别 1 和级别 2 打印机调整的 `Foomatic` PPD 文件对 `cups` 包中的通用 PPD 文件进行了补充：

- `/usr/share/cups/model/Postscript-level1.ppd.gz`
- `/usr/share/cups/model/Postscript-level2.ppd.gz`

cups-drivers 包中的 PPD 文件

通常，Foomatic 打印机过滤器 `foomatic-rip` 与非 PostScript 打印机的 Ghostscript 一起使用。合适的 Foomatic PPD 文件具有项“`*NickName: ... Foomatic/Ghostscript driver`”和“`*cupsFilter: ... foomatic-rip`”。这些 PPD 文件位于 `cups-drivers` 包中。

YaST 通常首选 `manufacturer-PPD` 文件。但是，如果没有适合的 `manufacturer-PPD` 文件，将选择带有项 `*NickName: ... Foomatic ... (recommended)` 的 Foomatic PPD 文件。

gutenprint 包中的 Gutenprint PPD 文件

Gutenprint（以前称为 GIMP-Print）中的 CUPS 过滤器 `rastertogutenprint`（而不是 `foomatic-rip`）可用于许多非 PostScript 打印机。`gutenprint` 包中提供了该过滤器和适合的 Gutenprint PPD 文件。Gutenprint PPD 文件位于 `/usr/share/cups/model/gutenprint/` 中并具有项 `*NickName: ... CUPS+Gutenprint` 和 `*cupsFilter: ... rastertogutenprint`。

manufacturer-PPDs 包中来自打印机制造商的 PPD 文件

`manufacturer-PPDs` 包中包含来自打印机制造商的 PPD 文件，这些文件是在充分自由的许可证下发布的。应该用打印机制造商的合适 PPD 文件配置 postscript 打印机，因为此文件支持使用 PostScript 打印机的所有功能。YaST 倾向于使用 `manufacturer-PPDs` 中的 PPD 文件。如果型号名称不匹配，则 YaST 不能使用 `manufacturer-PPDs` 包中的任何 PPD 文件。如果 `manufacturer-PPDs` 包对于相似型号（如 Funprinter 12xx 系列）仅包含一个 PPD 文件，则可能发生这种情况。在这种情况下，请手动在 YaST 中选择相应的 PPD 文件。

11.8 查错

下面几节介绍一些最常遇到的打印机硬件和软件问题以及解决或避免这些问题的方法。讨论的主题有 GDI 打印机、PPD 文件和端口配置。另外还讨论常见网络打印机问题、打印件问题以及队列处理。

11.8.1 打印机没有标准打印机语言支持

这些打印机不支持任何常见的打印机语言，只能使用专门的专有控制系列来进行寻址。因此这些打印机只能用于制造商提供了驱动程序的操作系统版本。GDI 是 Microsoft* 为图形设备开发的编程接口。通常制造商只提供 Windows 的驱动程序，而因为 Windows 驱动程序使用 GDI 界面，所有这些打印机也被称作 *GDI 打印机*。实际问题不是编程接口，而是这些打印机只能通过相应打印机型号的专用打印机语言进行处理。

某些 GDI 打印机可进行切换以 GDI 方式或一种标准打印机语言进行操作。请参见打印机手册看这是否可行。有些型号需要有专门的 Windows 软件来进行切换（注：Windows 打印机驱动程序在通过 Windows 进行打印时可能总是将打印机切换回 GDI 模式）。对于其他 GDI 打印机，还有针对标准打印机语言的扩展模块。

某些制造商为他们的打印机提供专有驱动程序。专有打印机驱动程序的缺点在于不能保证这些驱动程序可用于已安装的打印系统，也不能保证它们适合各种硬件平台。相反，支持标准打印机语言的打印机不依赖于特殊的打印系统版本或特殊的硬件平台。

与其花时间使专有 Linux 驱动程序工作，不如购买一台支持的打印机，这样更经济一些。这可以一次性全部解决驱动程序问题，从而无需安装并配置特殊驱动程序软件，也无需获取由于打印系统中开发的新功能而必须安装的驱动程序更新。

11.8.2 没有合适的 PPD 文件可用于 PostScript 打印机

如果 manufacturer-PPDs 包不包含任何用于 PostScript 打印机的合适 PPD 文件，则可以使用打印机制造商提供的驱动程序 CD 上的 PPD 文件或从打印机制造商 Web 网页下载合适的 PPD 文件。

如果以 zip 存档 (.zip) 或自解压缩 zip 存档 (.exe) 的形式提供 PPD 文件，则用 unzip 命令将其解包。首先，查看 PPD 文件的许可协议条款。然后使用 cupstestppd 实用程序来确认 PPD 文件是否与“Adobe PostScript 打印机描述文件格式规范 V4.3”相符合，如果实用程序返回“FAIL，”则描述 PPD 文件中的错误很严重，可能导致重大问题。应该解决 cupstestppd 报告的问题点。如果需要，询问打印机制造商是否提供合适的 PPD 文件。

11.8.3 并行端口

最安全的方法是将打印机直接连接到第一个并行端口并在 BIOS 中选择以下并行端口设置：

- I/O 地址：378（十六进制）
- 中断：无关
- 模式：Normal、SPP 或 Output Only
- DMA：禁用

如果即便进行了这些设置仍无法对并行端口上的打印机进行寻址，则按照 BIOS 中的设置在 /etc/modprobe.conf 中以 0x378 形式显式输入 I/O 地址。如果有两个并行端口，分别被设置为 I/O 地址 378 和 278（十六进制），则以 0x378,0x278 形式输入这两个端口。

如果中断 7 可用，则可以用 **例 11.1 “/etc/modprobe.conf：第一个并行端口的中断方式”** [118] 中显示的项将其激活。在激活中断方式之前，检查文件 /proc/interrupts 看看哪些中断仍在使用中。只显示当前正在使用的中断。根据哪些硬件部件处于活动状态，这可能会有所变化。用于并行端口的中断一

定不能被任何其他设备使用。如果您不确定，则使用巡回检测方式，设置 `irq=none`。

例 11.1 `/etc/modprobe.conf`: 第一个并行端口的中断方式

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=7
```

11.8.4 网络打印机连接

确定网络问题

将打印机直接连接到计算机。出于测试目的，将该打印机配置为本地打印机。如果打印机可以工作，则问题与网络有关。

检查 TCP/IP 网络

TCP/IP 网络和名称解析必须可以正常工作。

检查远程 lpd

使用以下命令测试是否可以与 `host` 上的 `lpd`（端口 515）建立 TCP 连接：

```
netcat -z host 515 && echo ok || echo failed
```

如果不能建立与 `lpd` 的连接，则 `lpd` 可能不处于活动状态或可能存在基本网络问题。

以 `root` 用户身份使用以下命令查询远程 `host` 上 `queue` 的状态报告（可能非常长），前提是相应的 `lpd` 处于活动状态并且主机接受查询：

```
echo -e "\004queue" \  
| netcat -w 2 -p 722 host 515
```

如果 `lpd` 不响应，则它可能不处于活动状态或可能存在基本网络问题。如果 `lpd` 响应，响应应该描述为什么在主机的队列上不能进行打印。如果您接收到类似 **例 11.2 “来自 `lpd` 的错误消息”** [118] 中的响应，则问题是由远程 `lpd` 引起的。

例 11.2 来自 `lpd` 的错误消息

```
lpd: your host does not have line printer access
lpd: queue does not exist
printer: spooling disabled
printer: printing disabled
```

检查远程 cupsd

默认情况下，CUPS 网络服务器应该每隔 30 秒在 UDP 端口 631 上广播其队列。因此，以下命令可用于测试网络中是否有 CUPS 网络服务器。执行此命令之前，务必停止本地 CUPS 守护程序。

```
netcat -u -l -p 631 & PID=$! ; sleep 40 ; kill $PID
```

如果广播 CUPS 网络服务器存在，则输出如例 11.3 “来自 CUPS 网络服务器的广播” [119]所示。

例 11.3 来自 CUPS 网络服务器的广播

```
ipp://192.168.2.202:631/printers/queue
```

以下命令可用于测试是否可以与 *host* 上的 cupsd（端口 631）建立 TCP 连接：

```
netcat -z host 631 && echo ok || echo failed
```

如果不能建立与 cupsd 的连接，则 cupsd 可能不处于活动状态或可能存在基本网络问题。如果 cupsd 处于活动状态并且主机接受查询，lpstat -h *host* -l -t 会返回 *host* 上所有队列的状态报告（可能非常长）。

下一个命令用于测试 *host* 上的 *queue* 是否接受由单个回车字符组成的打印作业。不应打印任何内容。可能会弹出一页空白纸。

```
echo -en "\r" \  
| lp -d queue -h host
```

对网络打印机或打印服务器计算机进行查错

当在打印服务器计算机中运行的假脱机程序要处理大量打印作业时，有时会导致出现问题。因为这是由打印服务器计算机中的假脱机程序引起的，所以没什么办法。作为替代解决方法，可以直接通过 TCP 套接字对连接到打印服务器计算机的打印机进行寻址来绕过打印服务器计算机中的假脱机程序。请参见第 11.4 节“网络打印机” [110]。

这样，打印服务器计算机仅用作数据传送（TCP/IP 网络和本地打印机连接）各种不同形式之间的转换器。要使用此方法，您需要知道打印服务器计算机上的 TCP 端口。如果打印机连接在打印服务器计算机上并且打开了电源，则通常可以在打开打印服务器计算机的电源后使用 nmap 包中的 nmap 实用程序确定此 TCP 端口。例如，nmap *IP-address* 可能会在打印服务器打印机中产生以下输出：

Port	State	Service
23/tcp	open	telnet
80/tcp	open	http
515/tcp	open	printer
631/tcp	open	cups
9100/tcp	open	jetdirect

此输出指出可以在端口 9100 上通过 TCP 套接字对连接到打印服务器计算机的打印机进行寻址。默认情况下，nmap 只检查在 /usr/share/nmap/nmap-services 中列出的一些常见的端口。要检查所有可能的端口，请使用命令 `nmap -p from_port-to_port IP-address`。这可能要花一些时间。有关详细信息，请参见 nmap 的手册页。

输入如下命令

```
echo -en "\rHello\r\f" | netcat -w 1 IP-address port
cat file | netcat -w 1 IP-address port
```

将字符串或文件直接发送到相应的端口以测试是否可以在该端口上对打印机进行寻址。

11.8.5 打印件有问题但没有错误消息

对于打印系统，打印作业完成的标志是 CUPS 后端完成到接收方（打印机）的数据传送。如果在接收方的进一步处理失败（例如，如果打印机无法打印打印机特定的数据），则打印系统不会对此进行通知。如果打印机无法打印打印机特定的数据，则选择另一个更适合该打印机的 PPD 文件。

11.8.6 禁用的队列

如果向接收方传送数据在多次尝试后都失败，则 CUPS 后端（例如 USB 或 socket）向打印系统（向 cupsd）报告一个错误。后端决定在报告数据传送无法完成之前是否继续尝试以及进行多少次尝试。由于继续尝试可能也是徒劳，cupsd 将禁用相应队列的打印。在消除了问题的起因后，系统管理员必须使用 `cupsenable` 命令重启用打印。

11.8.7 CUPS 浏览：删除打印作业

如果 CUPS 网络服务器通过浏览向客户机主机广播其队列并且客户机主机上合适的本地 cupsd 处于活动状态，则客户机 cupsd 接受来自应用程序的打印作业并将它们转发到服务器上的 cupsd。当 cupsd 接受打印作业后，会为该作业指派一个新的作业号。因此，客户机主机上的作业号与服务器上的作业号不同。因为通常都将打印作业立即转发，所以不能用客户机主机上的作业号将其删除，原因是一旦将打印作业转发到服务器 cupsd，客户机 cupsd 就会将打印作业视为已完成。

要删除服务器上的打印作业，请使用命令（例如 `lpstat -h cups.example.com -o`）确定服务器上的作业号（必须在服务器尚未完成该打印作业，即尚未完全将其发送到打印机的情况下）。使用此作业号，可以删除服务器上的打印作业：

```
cancel -h cups.example.com queue-jobnumber
```

11.8.8 有问题的打印作业和数据传送错误

如果打印进程中将打印机关闭或关闭计算机，则打印作业保留在队列中，当打开打印机或重引导计算机后，打印继续。必须使用 `cancel` 从队列中删除有问题的打印作业。

如果打印作业有问题或主机和打印机之间的通讯出现错误，则打印机会打印出很多张带有乱码的纸张，这是因为它不能正确处理数据。要解决此问题，请执行以下步骤：

- 1 要停止打印，请将所有纸张从喷墨打印机中取出或打开激光打印机的纸盒。高质量的打印机具有一个用于取消当前打印件的按钮。
- 2 打印作业可能仍在队列中，因为只有在将作业完全发送到打印机后才会将它们删除。使用 `lpstat -o` 或 `lpstat -h cups.example.com -o` 可以检查哪个队列当前正在打印。使用 `cancel queue-jobnumber` 或 `cancel -h cups.example.com queue-jobnumber` 可以删除打印作业。
- 3 即使已将打印作业从队列中删除，某些数据仍会被传送到打印机。检查 CUPS 后端进程是否仍在为相应的队列运行并将其终止。例如，对于连接

到并行端口的打印机，可以使用命令 `fuser -k /dev/lp0` 终止仍在访问打印机（更准确地说是并行端口）的所有进程。

- 4 通过关闭打印机一段时间完全重设置打印机。然后插入纸张并打开打印机。

11.8.9 对 CUPS 打印系统进行调试

使用以下通用过程确定 CUPS 打印系统中的问题：

- 1 在 `/etc/cups/cupsd.conf` 中设置 `LogLevel debug`。
- 2 停止 `cupsd`。
- 3 删除 `/var/log/cups/error_log*` 从而无需搜索非常长的日志文件。
- 4 启动 `cupsd`。
- 5 重复导致问题的操作。
- 6 检查 `/var/log/cups/error_log*` 中的消息以确定问题的原因。

11.8.10 更多信息

Novell 知识库 (<http://support.novell.com/>) 中提供了对许多特定问题的解决方案。通过对 CUPS 的文本搜索找到相关文章。

使用 udev 进行动态内核设备管理

12

内核几乎可以添加或删除运行系统中的任何设备。设备状态的更改（无论插入还是删除设备）需要通知用户空间。一旦插入或者发现设备时就需要进行配置。特定设备的用户需要知道此设备的所有状态更改。udev 提供必需的结构来动态维护设备节点文件以及 `/dev` 目录中的符号链接。udev 规则提供一种将外部设备插入到内核设备事件处理的方法。这使得您可以定制 udev 设备处理，例如通过添加特定脚本作为内核设备处理的一部分来执行，或者请求并导入额外数据从而在设备处理期间进行评估。

12.1 `/dev` 目录

`/dev` 目录中的设备节点提供对相应的内核设备的访问。使用 udev 时，`/dev` 目录反映内核的当前状态。每个内核设备都有相应的设备文件。如果设备从系统断开，则删除此设备节点。

`/dev` 目录的内容保存在临时文件系统中，所有文件都是在每个系统启动时从头创建的。手动创建或有意更改的文件不会在重引导后保留下来。无论相应内核设备的状态如何都出现在 `/dev` 目录中的静态文件和目录，可以放置在 `/lib/udev/devices` 目录中。系统启动时，此目录的内容复制到 `/dev` 目录，它们与 `/lib/udev/devices` 中的文件具有相同的所有权和许可权限。

12.2 内核 uevents 和 udev

必需的设备信息由 sysfs 文件系统导出。对于内核检测到并已初始化的设备，将创建一个带有该设备名称的目录。它包含带有特定于设备属性的属性文件。

每次添加或删除设备时，内核发送 `uevent` 来通知 `udev` 此情况。一旦启动后，`udev` 守护程序从 `/etc/udev/rules.d/*.rules` 文件读取并解析所有提供的规则并将它们保存在内存中。如果更改、添加或删除了规则文件，则守护程序可以使用命令 `udevadm control reload_rules` 重新装载所有规则在“记忆”中的表示形式。运行 `/etc/init.d/boot.udev reload` 时也会执行此操作。有关 `udev` 规则及其语法的更多细节，请参见第 12.6 节“使用 `udev` 规则影响内核设备事件处理” [126]。

每个接收到的事件都根据所提供的规则集进行匹配。这些规则可以增加或更改事件环境关键字、为要创建的设备节点请求特定名称、添加指向该节点的符号链接或者添加设备节点创建后运行的程序。从内核 `netlink` 套接字接收驱动程序内核 `uevent`。

12.3 驱动程序、内核模块和设备

设备的内核总线驱动程序探测。内核为每个检测到的设备创建内部设备结构，驱动程序内核将 `uevent` 发送到 `udev` 守护程序。总线设备通过特殊格式的 ID 来标识自己，这可以识别设备的类型。通常，这些 ID 由供应商和产品 ID 以及其他特定于子系统的值组成。每个总线都有自己对于这些 ID 的方案，称为 `MODALIAS`。内核获取设备信息，由此组成一个 `MODALIAS` ID 字符串，并将该字符串与事件一起发送。对于 USB 鼠标，如下所示：

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

每个设备驱动程序都带有它可以处理的设备的已知别名列表。这个列表包含在内核模块文件中。程序 `depmod` 读取 ID 列表并在内核的 `/lib/modules` 目录中为所有当前可用的模块创建文件 `modules.alias`。使用这种基础结构，模块的装载就如为每个带有 `MODALIAS` 关键字的事件调用 `modprobe` 一样简单。如果调用 `modprobe $MODALIAS`，它将组成该设备的设备别名与模块提供的别名相匹配。如果找到匹配的项，则装载该模块。这个通过 `udev` 触发并且自动发生。

12.4 引导和启动设备设置

在 `udev` 守护程序运行之前的引导过程中发生的所有设备事件都会丢失，因为处理这些事件的基础结构保存在 `root` 文件系统中，并且此时不可用。为了弥补此损失，内核提供了一个 `uevent` 文件，该文件位于 `sysfs` 文件系统每个设备的设备目录中。通过将 `add` 写入到该文件，内核将再次发送引导时丢失的相同事件。`/sys` 触发器中所有 `uevent` 文件的简单循环将再次触发所有事件来创建设备节点并执行设备设置。

例如，在引导期间出现的 USB 鼠标可能不会由早期引导逻辑初始化，因为驱动程序在那时不可用。此设备发现的事件丢失并且不能为该设备查找内核模块。不是手动搜索可能连接的设备，`udev` 在 `root` 文件系统可用后直接从内核请求所有设备事件，所以 USB 鼠标设备的事件可以再次运行。现在它在装入的 `root` 文件系统上找到内核模块，因此可以初始化 USB 鼠标。

在用户空间，设备冷插入序列和运行时期间发现的设备之间没有明显的区别。在这两种情况下，使用相同的规则来匹配并且运行相同的配置程序。

12.5 监视正在运行的 `udev` 守护程序

程序 `udevadm monitor` 可以用于将驱动程序核心事件和 `udev` 事件处理的计时可视化。

```
UEVENT[1185238505.276660] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
UDEV   [1185238505.279198] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
UEVENT[1185238505.279527] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UDEV   [1185238505.285573] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UEVENT[1185238505.298878] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UDEV   [1185238505.305026] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UEVENT[1185238505.305442] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
UEVENT[1185238505.306440] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV   [1185238505.325384] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV   [1185238505.342257] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
```

UEVENT 行显示内核已经通过 **netlink** 发送的事件。UDEV 行显示已经完成的 **udev** 事件处理程序。计时以微秒为单位显示。UEVENT 和 UDEV 之间的时间是 **udev** 用于处理此事件或者 **udev** 守护程序延迟执行从而同步此事件与相关以及已运行的事件的时间。例如，硬盘分区的事件总是等待主磁盘设备事件完成，因为分区事件可能依赖主磁盘事件从硬件查询的数据。

`udevadm monitor --env` 显示完整的事件环境：

```
ACTION=add
DEVPATH=/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10
SUBSYSTEM=input
SEQNUM=1181
NAME="Logitech USB-PS/2 Optical Mouse"
PHYS="usb-0000:00:1d.2-1/input0"
UNIQ=""
EV=7
KEY=70000 0 0 0 0
REL=103
MODALIAS=input:b0003v046DpC03Ee0110-e0,1,2,k110,111,112,r0,1,8,amlsfw
```

udev 也将消息发送给 **syslog**。用于控制将哪些消息发送到 **syslog** 的默认 **syslog** 优先权在 **udev** 配置文件 `/etc/udev/udev.conf` 中指定。可以使用 `udevadm control log_priority=level/number` 更改正在运行的守护程序的日志优先权。

12.6 使用 **udev** 规则影响内核设备事件处理

udev 规则可以与内核添加到事件本身的属性或者内核导出到 **sysfs** 的任何信息相匹配。规则还可以从外部程序请求其他信息。根据提供的规则匹配每个事件。所有规则都位于 `/etc/udev/rules.d` 目录下。

规则文件中的每一行至少包含一个关键字值对。有两种类型的关键字，匹配关键字和指派关键字。如果所有匹配关键字与它们的值匹配，则应用此规则并将指派关键字指派给特定的值。匹配规则可以指定设备节点的名称、将符号链接指向该节点或者运行特定程序作为事件处理的一部分。如果找不到匹配的规则，则使用默认设备节点名来创建设备节点。在 **udev** 手册页中描述了有关规则语法和提供用来匹配或导入数据的关键字的详细信息。下面的样本规则提供了 **udev** 规则语法的基本介绍。这些样本规则全部取自 `/etc/udev/rules.d/50-udev-default.rules` 下的 **udev** 默认规则集。

例 12.1 样本 udev 规则

```
# console
KERNEL=="console", MODE="0600", OPTIONS="last_rule"

# serial devices
KERNEL=="ttyUSB*", ATTRS{product}=="[Pp]alm*Handheld*", SYMLINK+="pilot"

# printer
SUBSYSTEM=="usb", KERNEL=="lp*", NAME="usb/%k", SYMLINK+="usb%k", GROUP="lp"

# kernel firmware loader
SUBSYSTEM=="firmware", ACTION=="add", RUN+="firmware.sh"
```

console 规则由三个关键字构成：一个匹配关键字 (KERNEL)，两个赋值关键字 (MODE、OPTIONS)。KERNEL 匹配规则搜索设备列表以查找类型为 console 的所有项。只有完全匹配才有效，才能触发执行此规则。在这种情况下，MODE 关键字为设备节点指派特殊权限，仅为该设备的拥有者指派读写权限。OPTIONS 关键字将该规则标记为此类型的所有设备最后采用的规则。匹配此特殊设备类型的任何后续规则都不产生任何影响。

50-udev-default.rules 中不再提供 serial devices 规则，但该规则仍然值得注意。该规则由两个匹配关键字 (KERNEL 和 ATTRS) 和一个赋值关键字 (SYMLINK) 构成。KERNEL 关键字搜索类型为 ttyUSB 的所有设备。该关键字使用 * 通配符匹配这些设备中的几个。第二个匹配关键字 ATTRS 检查任何 ttyUSB 设备的 sysfs 中的 product 属性文件是否包含特定字符串。赋值关键字 (SYMLINK) 将符号链接添加至该设备的 /dev/pilot 下。此关键字中使用的运算符 (+=) 告知 udev 进一步执行此操作，即使前面或后面的规则添加其他符号链接。由于此规则包含两个匹配关键字，因此仅当两个条件都满足时，才应用。

printer 规则处理 USB 打印机，其中包含两个匹配关键字 (SUBSYSTEM 和 KERNEL)，并且必须同时应用这两个关键字，才能应用整个规则。三个赋值关键字处理该设备类型的命名 (NAME)、符号设备链接 (SYMLINK) 的创建，以及此设备类型的组成员资格 (GROUP)。在 KERNEL 关键字中使用通配符 * 将使其匹配若干 lp 打印机设备。NAME 和 SYMLINK 关键字中都使用了替换项，以便按内部设备名称扩展这些字符串。例如，指向第一个 lp USB 打印机的符号链接为 /dev/usblp0。

kernel firmware loader 规则用于使 udev 在运行时期间通过外部助手脚本本装载其他固件。SUBSYSTEM 匹配关键字搜索 firmware 子系统。ACTION

关键字检查是否添加了属于 `firmware` 子系统的任何设备。`RUN+=` 关键字触发执行 `firmware.sh` 脚本，以便找到应装载的固件。

所有规则具有一些共同的特征：

- 每个规则由一个或多个以逗号分隔的关键字值对构成。
- 关键字的运算由运算符确定。`udev` 规则支持多个不同的运算符。
- 每个给定值必须用引号引起来。
- 规则文件的每一行代表一个规则。如果一个规则超过一行，请使用 \ 合并不同行，就像在壳层语法中一样。
- `udev` 规则支持匹配 `*`、`?` 和 `[]` 模式的壳层式模式。
- `udev` 规则支持替换。

12.6.1 在 `udev` 规则中使用运算符

创建可以从若干不同运算符选择的关键字，具体取决于希望创建的关键字类型。匹配关键字通常仅用于查找匹配或明显不匹配搜索值的值。匹配关键字包含以下运算符之一：

`==`

比较等于性。如果关键字包含搜索模式，则匹配该模式的所有结果均有效。

`!=`

比较不等于性。如果关键字包含搜索模式，则匹配该模式的所有结果均有效。

赋值关键字可以使用下面的任何运算符：

`=`

为关键字指派值。如果关键字以前由一系列值构成，关键字将重置，并且仅指派一个值。

`+=`

为包含一系列项的关键字添加一个值。

`:=`

指派最终值。不允许后面的规则进行任何后续更改。

12.6.2 在 udev 规则中使用替换项

udev 规则支持使用占位符和替换项。请按照在其他任何脚本中的相同方式使用。在 udev 规则中可使用以下替换项：

`%r`、`$root`

设备目录 `/dev`（默认）。

`%p`、`$devpath`

`DEVPATH` 的值。

`%k`、`$kernel`

`KERNEL` 的值或内部设备名称。

`%n`、`$number`

设备号。

`%N`、`$tempnode`

设备文件的临时名称。

`%M`、`$major`

设备的主编号。

`%m`、`$minor`

设备的次编号。

`%s{attribute}/$attr{attribute}`

`sysfs` 属性的值（由 `attribute` 指定）。

`%E{variable}`、`$attr{variable}`

环境变量的值（由 `variable` 指定）。

`%c`、`$result`

`PROGRAM` 的输出。

%%

% 字符。

\$\$

\$ 字符。

12.6.3 使用 udev 匹配关键字

匹配关键字描述应用 udev 规则之前必须满足的条件。以下匹配关键字可用：

ACTION

事件操作的名称，如 add 或 remove（添加或删除设备时）。

DEVPATH

事件设备的设备路径，如 DEVPATH=/bus/pci/drivers/ipw3945，用于搜索与 ipw3945 驱动程序有关的所有事件。

KERNEL

事件设备的内部（内核）名称。

SUBSYSTEM

事件设备的子系统，如 SUBSYSTEM=usb（用于与 USB 设备有关的所有事件）。

ATTR{*filename*}

事件设备的 sysfs 属性。例如，要匹配 vendor 属性文件名中包含的字符串，可以使用 ATTR{vendor}=="On[ss]tream"。

KERNELS

让 udev 向上搜索设备路径以查找匹配的设备名称。

SUBSYSTEMS

让 udev 向上搜索设备路径以查找匹配的设备子系统名称。

DRIVERS

让 udev 向上搜索设备路径以查找匹配的设备驱动程序名称。

ATTRS{*filename*}

让 udev 向上搜索设备路径以查找具有匹配的 sysfs 属性值的设备。

ENV{key}

环境变量的值，如ENV{ID_BUS}="ieee1394，用于搜索与该FireWire总线ID有关的所有事件。

PROGRAM

让udev执行外部程序。程序必须返回退出码零，才能成功。程序的输出（打印到 stdout）可用于 RESULT 关键字。

RESULT

匹配上次 PROGRAM 调用的输出字符串。在与 PROGRAM 关键字相同的规则中包含该关键字，或在后面的一个中。

12.6.4 使用 udev 赋值关键字

与上述匹配关键字相比，赋值关键字不描述必须满足的条件，而是为 udev 维护的设备节点指派值、名称和操作。

名称

将创建的设备节点的名称。在一个规则设置节点名称之后，将对该节点忽略带有 NAME 关键字的其他所有规则。

SYMLINK

与要创建的节点有关的符号链接名称。多个匹配的规则可添加要使用设备节点创建的符号链接。也可以通过使用空格字符分隔符号链接名称，在一个规则中为一个节点指定多个符号链接。

OWNER, GROUP, MODE

新设备节点的权限。此处指定的值重写已编译的任何值。

ATTR{key}

指定要写入事件设备的 sysfs 属性的值。如果使用 == 运算符，也将使用该关键字匹配 sysfs 属性的值。

ENV{key}

告知 udev 将变量导出到环境。如果使用 == 运算符，也将使用该关键字匹配环境变量。

RUN

告知 **udev** 向程序列表添加要为该设备执行的程序。请注意，将此操作限制于很短的任务，以免妨碍此设备的后续事件。

LABEL

添加 GOTO 可跳至的标签。

GOTO

告知 **udev** 跳过一些规则，继续执行具有按 GOTO 关键字引用的标签的规则。

IMPORT{type}

将变量装载入外部程序输出之类的事件环境中。**udev** 导入不同类型的若干变量。如果未指定任何类型，**udev** 将尝试根据文件权限的可执行位来自行确定类型。

- `program` 告知 **udev** 执行外部程序并导入其输出。
- `file` 告知 **udev** 导入文本文件。
- `parent` 告知 **udev** 从父设备导入储存的关键字。

WAIT_FOR_SYSFS

告知 **udev** 等待要为某个设备创建的指定 `sysfs` 文件（如

`WAIT_FOR_SYSFS="ioerr_cnt"`）通知 **udev** 等至此 `ioerr_cnt` 文件创建完成。

选项

OPTION 关键字可能有若干值：

- `last_rule` 告知 **udev** 忽略后面的所有规则。
- `ignore_device` 告知 **udev** 完全忽略此事件。
- `ignore_remove` 告知 **udev** 忽略后面针对设备的所有删除事件。
- `all_partitions` 告知 **udev** 为块设备上的所有可用分区创建设备节点。

12.7 永久设备命名

动态设备目录和 `udev` 规则基础结构可以为所有磁盘设备提供固定名称，而不考虑它们的识别顺序或设备使用的连接。内核创建的每个相应的块设备由工具根据有关特定总线、驱动器类型或者文件系统的特殊知识进行检查。除了动态内核提供的设备节点名，`udev` 还保留各种指向该设备的永久符号链接：

```
/dev/disk
|-- by-id
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B -> ../../sda
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part1 -> ../../sda1
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part6 -> ../../sda6
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part7 -> ../../sda7
|   |-- usb-Generic_STORAGE_DEVICE_02773 -> ../../sdd
|   `-- usb-Generic_STORAGE_DEVICE_02773-part1 -> ../../sdd1
|-- by-label
|   |-- Photos -> ../../sdd1
|   |-- SUSE10 -> ../../sda7
|   `-- devel -> ../../sda6
|-- by-path
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0 -> ../../sda
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part1 -> ../../sda1
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part6 -> ../../sda6
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part7 -> ../../sda7
|   |-- pci-0000:00:1f.2-scsi-1:0:0:0 -> ../../sr0
|   |-- usb-02773:0:0:2 -> ../../sdd
|   |-- usb-02773:0:0:2-part1 -> ../../sdd1
`-- by-uuid
    |-- 159a47a4-e6e6-40be-a757-a629991479ae -> ../../sda7
    |-- 3e999973-00c9-4917-9442-b7633bd95b9e -> ../../sda6
    `-- 4210-8F8C -> ../../sdd1
```

12.8 udev 使用的文件

`/sys/*`

Linux 内核提供的虚拟文件系统，用于导出所有当前已知设备。此信息由 `udev` 用于在 `/dev` 中创建设备节点

`/dev/*`

动态创建的设备节点和引导时从 `/lib/udev/devices/*` 复制的静态内容

以下文件和目录包含 `udev` 基础结构的关键元素：

`/etc/udev/udev.conf`

主 `udev` 配置文件。

`/etc/udev/rules.d/*`

`udev` 事件匹配规则。

`/lib/udev/devices/*`

静态 `/dev` 内容。

`/lib/udev/*`

从 `udev` 规则调用的帮助程序。

12.9 有关详细信息

有关 `udev` 基础结构的更多信息，请参见以下手册页：

`udev`

有关 `udev`、关键字、规则和其他重要配置问题的常规信息。

`udevadm`

`udevadm` 可用于控制 `udev` 的运行时行为，请求内核事件，管理事件队列，以及提供简单的调试机制。

`udev`

有关 `udev` 事件管理守护程序的信息。

X Window 系统

X Window 系统 (X11) 是 UNIX 中图形用户界面的实际标准。X 是基于网络的，可以使在一个主机上启动的应用程序显示在通过任何类型的网络（LAN 或 Internet）连接的另一个主机上。本章介绍了 X Window 系统环境的安装和优化，并提供了关于在 SUSE® Linux Enterprise Desktop 中使用字体的背景信息。

13.1 手动配置 X Window 系统

默认设置下，以第 6.2 节 “Setting Up Graphics Card and Monitor” (第 6 章 *Setting Up Hardware Components with YaST*, ↑ *部署指南*) 中所述用 SaX2 界面配置 X Window 系统。或者也可以通过编辑其配置文件手动配置它。

警告: 错误的 X 配置可能会损坏您的硬件。

配置 X Window 系统时要小心。在完成配置前，切勿启动 X Window 系统。错误配置的系统可能会对您的硬件造成无法修复的损坏（此情况尤其针对于固定频率的监视器）。该书和 SUSE Linux Enterprise Desktop 的创建者不对导致的任何损坏负责。这里提供的信息已经仔细斟酌，但不能保证所提供的所有方法均正确且不会对您的硬件造成任何损坏。

命令 `sax2` 会创建 `/etc/X11/xorg.conf` 文件。这是 X Window 系统的主配置文件。请在此查找与图形卡、鼠标和监视器有关的所有设置。

重要: 使用 X -configure

使用 `x -configure` 配置您的 X 安装（如果之前尝试 **SUSE Linux Enterprise Desktop** 的 **SaX2** 失败）。如果您的安装涉及专用的仅二进制驱动程序，`x -configure` 不起作用。

下面小节介绍配置文件 `/etc/X11/xorg.conf` 的结构。它由多个部分组成，每个部分处理配置的某个特定方面。每个部分都以关键字 `Section` `<designation>` 开头，以 `EndSection` 结尾。以下惯例适用于所有章节：

```
Section "designation"
    entry 1
    entry 2
    entry n
EndSection
```

表 13.1 “`/etc/X11/xorg.conf` 中的部分” [136]中列出了可用的部分类型。

表 13.1 `/etc/X11/xorg.conf` 中的部分

类型	含义
文件	用于字体和 RGB 颜色表的路径。
ServerFlags	服务器行为的常规切换。
模块	服务器应装载的模块列表
InputDevice	此部分配置输入设备，例如键盘和特殊输入设备（触摸板、游戏杆等）。此部分的重要参数有 <code>Driver</code> 以及定义 <code>Protocol</code> 和 <code>Device</code> 的选项。对连接到计算机的每个设备，通常都有一个 <code>InputDevice</code> 部分。
监视程序	使用的显示器。此部分的重要元素是标识符（稍后在 <code>Screen</code> 定义中引用）、刷新率 <code>VertRefresh</code> 和同步频率限制（ <code>Horizsync</code> 和 <code>VertRefresh</code> ）。这些设置采用的单位为 <code>MHz</code> 、 <code>kHz</code> 和 <code>Hz</code> 。通常，服务器拒绝不符合监视器规格的任何方式行。这样可防止意外地将过高的频率发送到监视器。

类型	含义
方式	<p>特定屏幕分辨率的方式行参数。可以根据用户给出的值由 SaX2 计算出这些参数，并且通常无需更改这些参数。您可以在此时进行手动干预，例如当要连接固定频率监视器时。</p> <p>HOWTO 文件（位于 <code>/usr/share/doc/howto/en/html/XFree86-Video-Timings-HOWTO</code>）提供了各个数字值含义的细节（在 <code>howtoenh</code> 包中提供）。要手动计算 VESA 方式，可使用工具 <code>cvt</code>。例如，要计算 1680x1050@60Hz 监视器的模式行，请使用命令 <code>cvt 1680 1050 60</code>。</p>
设备	<p>特定的图形卡。系统通过其描述性名称来参考图形卡。本部分中的可用选项高度依赖于所用驱动程序。例如，如果使用 <code>i810</code> 驱动程序，请在手册页 <code>man 4 i810</code> 中查找有关各种可用选项的更多信息。</p>
屏幕	<p>将 <code>Monitor</code> 和 <code>Device</code> 放在一起以组成 X.Org 的所有必要设置。在 <code>Display</code> 子部分中，指定虚拟屏幕 (<code>Virtual</code>) 的大小、<code>ViewPort</code> 以及此屏幕所用的 <code>Modes</code>。</p> <p>请注意，某些驱动程序要求 <code>Display</code> 部分中某个位置内必须存在所用的全部配置。例如，如果使用便携式计算机并希望使用比内部 LCD 更大的外部监视器，可能需要在 <code>Modes</code> 行结尾处添加内部 LCD 支持的更大分辨率。</p>
<code>ServerLayout</code>	<p>单个或多头配置的布局。此部分将输入设备 <code>InputDevice</code> 和显示设备 <code>Screen</code> 绑定在一起。</p>
<code>DRI</code>	<p>提供 Direct Rendering Infrastructure (DRI) 的信息。</p>

下面详细介绍 `Monitor`、`Device` 和 `Screen`。`X.Org` 和 `xorg.conf` 的手册页提供了有关其他部分的详细信息。

`xorg.conf` 中可以存在多个不同的 `Monitor` 和 `Device` 部分。甚至可以存在多个 `Screen` 部分。`ServerLayout` 部分确定使用其中哪个部分。

13.1.1 Screen 部分

Screen 部分将 Monitor 部分与 Device 部分结合起来并确定要使用的分辨率和颜色深度。Screen 部分与例 13.1 “文件 /etc/X11/xorg.conf 的 Screen 部分” [138] 类似。

例 13.1 文件 /etc/X11/xorg.conf 的 Screen 部分

```
Section "Screen"❶
    DefaultDepth 16❷
    SubSection "Display"❸
        Depth 16❹
        Modes "1152x864" "1024x768" "800x600"❺
        Virtual 1152x864❻
    EndSubSection
    SubSection "Display"
        Depth 24
        Modes "1280x1024"
    EndSubSection
    SubSection "Display"
        Depth 32
        Modes "640x480"
    EndSubSection
    SubSection "Display"
        Depth 8
        Modes "1280x1024"
    EndSubSection
    Device "Device[0]"
    Identifier "Screen[0]"❼
    Monitor "Monitor[0]"
EndSection
```

- ❶ Section 确定该部分的类型，在本示例中是 Screen。
- ❷ DefaultDepth 决定默认使用的颜色深度（除非明确指定其他颜色深度）。
- ❸ 对每种颜色深度指定不同的 Display 子部分。
- ❹ Depth 决定对本组 Display 设置使用的颜色深度。可用值有 8、15、16、24 和 32，尽管并非所有 X 服务器模块或分辨率都可以支持所有这些值。
- ❺ Modes 部分由可能的屏幕分辨率列表组成。X 服务器从左到右检查此列表。对于每个分辨率，X 服务器均会在 Modes 部分中搜索合适的 Modeline。Modeline 取决于监视器和图形卡的功能。Monitor 设置确定最终的 Modeline。

找到的第一个分辨率是 Default mode。使用 **Ctrl+Alt++**（在数字小键盘上）向右切换到列表中的下一个分辨率。使用 **Ctrl+Alt+-**（在数字小键盘上）切换到上一个。这使您能够在 X 运行时改动分辨率。

- ⑥ 包含 Depth 16 的 Display 子部分的最后一行指出了虚拟屏幕的大小。虚拟屏幕的最大可能大小取决于图形卡中安装的内存量和所需的颜色深度，而不取决于监视器的最大分辨率。如果忽略此行，虚拟分辨率就是物理分辨率。因为目前的图形卡都具有大量视频内存，所以您可以创建非常大的虚拟桌面。但是，如果您将大部分视频内存用于虚拟桌面，则可能不能再使用 3D 功能。例如，如果图形卡有 16 MB 视频 RAM，则当采用 8 位颜色深度时，虚拟屏幕最多可以有 4096x4096 个像素。但建议不要将所有内存用于虚拟屏幕，因为图形卡的内存还要用于多种字体和图形缓存，对于加速卡而言尤其如此。
- ⑦ 行 Identifier（这里是 Screen[0]）为此部分指定一个定义的名称，在随后的 ServerLayout 部分中可以使用此名称唯一引用这个部分。行 Device 和 Monitor 指定属于此定义的图形卡和监视器。这些行仅仅是通过 Device 和 Monitor 部分的相应名称或标识符指向这些部分的链接。下面详细讨论这些部分。

13.1.2 Device 部分

Device 部分描述特定的图形卡。您可以在 `xorg.conf` 中包含任意多个设备项，前提是要使用关键字 Identifier 对这些项的名称进行区分。如果您安装了多个图形卡，通常按顺序对这些部分进行编号。第一个设备称为 Device[0]，第二个设备称为 Device[1]，依此类推。以下文件是从安装有 Matrox Millennium PCI 图形卡（由 SaX2 配置）的计算机的 device 部分摘出的一段：

```
Section "Device"
    BoardName      "MGA2064W"
    BusID          "0:19:0"❶
    Driver         "mga"❷
    Identifier     "Device[0]"
    VendorName     "Matrox"
    Option         "sw_cursor"
EndSection
```

- ❶ BusID 是指安装图形卡的 PCI 或 AGP 插槽。它与使用命令 `lspci` 显示的 ID 相匹配。X 服务器需要采用十进制形式的详细信息，但 `lspci` 以十六进制形式显示这些信息。BusID 的值由 SaX2 自动检测。

- ② `driver` 的值由 **SaX2** 自动设置，指定哪个驱动程序用于您的图形卡。如果此卡是 **Matrox Millennium**，则将驱动程序模块称为 `mga`。然后，**X** 服务器通过 `drivers` 子目录的 `Files` 部分中定义的 `ModulePath` 进行搜索。在标准安装中，有一个 `/usr/lib/xorg/modules/drivers` 目录或 `/usr/lib64/xorg/modules/drivers` 目录，用于 64 位操作系统目录。然后将 `_drv.o` 添加到名称中。因此，对于 `mga` 驱动程序，将装载驱动程序文件 `mga_drv.o`。

还可以通过其他选项影响 **X** 服务器或驱动程序的行为。在 `Device` 部分中设置的选项 `sw_cursor` 就是这方面的一个示例。此选项取消激活硬件鼠标光标并使用软件显示鼠标光标。根据驱动程序模块，有不同的选项可用，它们位于目录 `/usr/share/doc/package_name` 中驱动程序模块的描述文件中。通常还可以在手册页 (`man xorg.conf`、`man 4<driver module>` 和 `man 4 chips`) 中查看有效的选项。

如果图形卡有多个视频连接器，可以将这一个卡的不同设备配置为单一视图。使用 **SaX2** 以这种方式对图形接口进行设置。

13.1.3 Monitor 部分和 Modes 部分

与 `Device` 部分类似，`Monitor` 和 `Modes` 部分分别描述一个监视器。配置文件 `/etc/X11/xorg.conf` 可以包含任意多个 `Monitor` 部分。每个 `Monitor` 部分使用行 `UseModes`（如果可用）引用一个 `Modes` 部分。如果没有 `Modes` 部分可用于 `Monitor` 部分，**X** 服务器将根据常规同步值计算相应值。服务器布局部分指定相关的 `Monitor` 部分。

只有有经验的用户才可以设置监视器定义。`modeline` 是 `Monitor` 部分的重要组成部分。方式行设置相应分辨率的水平定时和垂直定时。`Monitor` 部分储存有监视器属性（特别是所允许的频率）。可以使用实用程序 `cvt` 生成 **VESA** 方式。有关更多信息，请参见 `cvt man cvt` 的手册页。

警告

除非您对监视器和图形卡功能有深入了解，否则建议不要更改 `modelien`，因为这可能严重损坏监视器。

如果您要创建自己的监视器描述，则应非常熟悉 `/usr/share/X11/doc` 中的文档。安装包 `xorg-x11-doc` 以查找 PDF 和 HTML 页面。

现在，很少需要手动指定方式行。如果您使用的是最新的多频同步监视器，则通常由 `x` 服务器通过 DDC 直接从监视器中读取允许的频率和最佳分辨率，如 [SaX2 配置一节](#) 所述。如果由于某种原因无法执行此操作，请使用 X 服务器中包含的 VESA 方式之一。这种方式可用于大多数图形卡和监视器的组合。

13.2 安装和配置字体

在 SUSE Linux Enterprise Desktop 中安装附加字体非常简单。只需要将字体复制到位于 X11 字体路径中的任何目录即可（请参见 [第 13.2.1 节“X11 核心字体”](#) [142]）。要启用字体，安装目录应是 `/etc/fonts/fonts.conf` 中配置的目录的子目录（请参见 [第 13.2.2 节“Xft”](#) [143]），或用 `/etc/fonts/suse-font-dirs.conf` 包含到此文件中。

以下是 `/etc/fonts/fonts.conf` 中的摘录。该文件是标准的配置文件，应适合大多数配置。它还定义包含的目录 `/etc/fonts/conf.d`。在此目录中，以两位数字开头的所有文件或符号链接均由 `fontconfig` 装载。有关此功能的更详细描述，请参见 `/etc/fonts/conf.d/README`。

```
<!-- Font directory list -->
<dir>/usr/share/fonts</dir>
<dir>/usr/X11R6/lib/X11/fonts</dir>
<dir>/opt/kde3/share/fonts</dir>
<dir>/usr/local/share/fonts</dir>
<dir>~/.fonts</dir>
<include ignore_missing="yes">conf.d</include>
```

`/etc/fonts/suse-font-dirs.conf` 会自动生成，以引入随（多为第三方）应用程序附送的字体，如 [OpenOffice.org](#)、Java 或 [Adobe Acrobat Reader](#)。`/etc/fonts/suse-font-dirs.conf` 的典型条目外观如下：

```
<dir>/usr/lib64/ooo-2.0/share/fonts</dir>
<dir>/usr/lib/jvm/java-1_4_2-sun-1.4.2.11/jre/lib/fonts</dir>
<dir>/usr/lib64/jvm/java-1.5.0-sun-1.5.0_07/jre/lib/fonts</dir>
<dir>/usr/X11R6/lib/Acrobat7/Resource/Font</dir>
<dir>/usr/X11R6/lib/Acrobat7/Resource/Font/PFM</dir>
```

要在整个系统安装其他字体，请手动将字体文件复制至适当的目录（如root），例如 `/usr/share/fonts/truetype`。或者，可以使用 KDE 控制中心中的 KDE 字体安装程序来执行此任务。结果是一样的。

您还可以创建符号链接，而不复制实际字体。例如，如果已装入的 Windows 分区上的字体已获得许可并要使用，则可能要执行此操作。随后，运行 `SuSEconfig --module fonts`。

`SuSEconfig --module fonts` 执行脚本 `/usr/sbin/fonts-config`，该脚本处理字体的配置。有关此脚本的更多信息，请参见其手册页 (`man fonts-config`)。

上面的过程同样适用于位图字体、TrueType 和 OpenType 字体以及 Type1 (PostScript) 字体。可以将所有这些字体类型安装在任何目录中。

X.Org 包含两个完全不同的字体系统：旧的 *x11* 核心字体系统和新设计的 *Xft* 和 *fontconfig* 系统。下面几节简要介绍这两种系统。

13.2.1 X11 核心字体

目前，X11 核心字体系统不仅支持位图字体，还支持可缩放字体（例如 Type1 字体）、TrueType 以及 OpenType 字体。X11 核心字体系统只在没有反锯齿处理和子像素显示的情况下支持可缩放字体，并且装载许多语言具有字形的大型可缩放字体可能需要较长的时间。也支持 Unicode 字体，但使用它们的速度比较慢，而且需要更多内存。

X11 核心字体系统带有一些固有缺陷。它已经过时，而且不再能以有意义的方式扩展。虽然为了实现向后兼容而不得不保留 X11 核心字体系统，但应尽可能使用更先进的 *Xft* 和 *fontconfig* 系统。

为了执行相应的操作，X 服务器需要知道它可使用的字体以及在系统中的哪些位置可找到这些字体。这由 `FontPath` 变量来处理，该变量包含所有有效系统字体目录的路径。在其中每个目录中，一个名为 `fonts.dir` 的文件会列出此目录中的可用字体。`FontPath` 由 X 服务器在启动时生成。它将在配置文件 `/etc/X11/xorg.conf` 的每个 `FontPath` 项中搜索有效的 `fonts.dir` 文件。这些项位于 `Files` 部分。使用 `xset q` 可显示实际的 `FontPath`。运行时也可以使用 `xset` 更改该路径。要添加其他路径，请使用 `xset+fp <path>`。要删除不需要的路径，请使用 `xset-fp <path>`。

如果 X 服务器已经处于活动状态，则可以使用命令 `xsetfp rehash` 使装入的目录中新安装的字体可用。通过 `SuSEconfig--module fonts` 执行此命令。因为命令 `xset` 需要访问正在运行的 X 服务器，所以只有当从可以访问正在运行的 X 服务器的 shell 启动 `SuSEconfig--module fonts` 时，此命令才能发挥作用。实现此操作最简单的方法是通过输入 `su` 和 `root` 密码获得 `root` 权限。`su` 会将启动 X 服务器的用户的访问权限转移到 `root` 外壳。要检查是否正确安装了字体以及是否可以通过 X11 核心字体系统使用字体，请使用命令 `xlsfonts` 列出所有可用字体。

默认情况下，SUSE Linux Enterprise Desktop 使用 UTF-8 区域设置。因此，应首选 Unicode 字体（`xlsfonts` 输出中以 `iso10646-1` 结尾的字体名称）。可以使用 `xlsfonts | grep iso10646-1` 列出所有可用的 Unicode 字体。几乎所有在 SUSE Linux Enterprise Desktop 中可用的 Unicode 字体都至少包括欧洲语言所需的字形（以前编码为 `iso-8859-*`）。

13.2.2 Xft

从一开始，Xft 的编程人员就确保该系统可以很好地支持可缩放字体（包括反锯齿处理）。如果使用 Xft，则是由使用字体的应用程序显示字体，而不是像 X11 核心字体系统中由 X 服务器显示字体。采用这种方式，相应的应用程序能够访问实际字体文件并完全控制如何显示字形。这就为正确显示多种语言的文本奠定了基础。直接访问字体文件对于用于打印的嵌入字体非常有用，因为这样可以确保打印输出与屏幕输出看上去完全一样。

在 SUSE Linux Enterprise Desktop 中，两个桌面环境 KDE 和 GNOME、Mozilla 及许多其他应用程序均已默认使用 Xft。使用 Xft 的应用程序在数目上已经超过了使用以前的 X11 核心字体系统的应用程序。

Xft 使用 `fontconfig` 库来查找字体并影响字体的显示方式。`fontconfig` 的属性由全局配置文件 `/etc/fonts/fonts.conf` 控制。应向 `/etc/fonts/local.conf` 和用户特定的配置文件 `~/.fonts.conf` 添加特殊配置。所有这些 `fontconfig` 配置文件的开头必须是

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

并且结尾必须是

```
</fontconfig>
```

要添加用于搜索字体的目录，请追加类似下面内容的一行：

```
<dir>/usr/local/share/fonts/</dir>
```

但通常没有必要这样做。默认情况下，已经在 `/etc/fonts/fonts.conf` 中输入了用户特定的目录 `~/.fonts`。因此，要安装附加字体，只需将它们复制到 `~/.fonts` 即可。

您还可以插入用来确定字体外观的规则。例如，输入

```
<match target="font">
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

来禁用所有字体的反锯齿处理，或输入

```
<match target="font">
  <test name="family">
    <string>Luxi Mono</string>
    <string>Luxi Sans</string>
  </test>
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

来禁用特定字体的反锯齿处理。

默认情况下，大多数应用程序使用字体名称 `sans-serif`（或等效的 `sans`）、`serif` 或 `monospace`。它们不是真正的字体，而只是可解析为合适的字体（取决于语言设置）的别名。

用户可以方便地将规则添加到 `~/.fonts.conf` 中，以将这些别名解析为他们喜欢的字体：

```
<alias>
  <family>sans-serif</family>
  <prefer>
    <family>FreeSans</family>
  </prefer>
</alias>
<alias>
  <family>serif</family>
  <prefer>
    <family>FreeSerif</family>
  </prefer>
```

```
</alias>
<alias>
  <family>monospace</family>
  <prefer>
    <family>FreeMono</family>
  </prefer>
</alias>
```

因为几乎所有应用程序都默认使用这些别名，所以这几乎影响到整个系统。这样，您可以方便地在几乎所有位置都使用自己喜欢的字体，而无需在各个应用程序中修改字体设置。

使用 `fc-list` 命令可以查看已安装了哪些字体以及哪些字体可用。例如，命令 `fc-list` 返回所有字体的列表。要查看可用的可缩放字体 (`:scalable=true`) 中有哪些包含希伯来语 (`:lang=he`) 所需的所有字形、它们的字体名称 (`family`)、字型 (`style`)、粗细 (`weight`) 以及包含这些字体的文件的名称，请输入以下命令：

```
fc-list ":lang=he:scalable=true" family style weight
```

此命令的输出类似于下面：

```
Lucida Sans:style=Demibold:weight=200
DejaVu Sans:style=Bold Oblique:weight=200
Lucida Sans Typewriter:style=Bold:weight=200
FreeSerif:style=Bold,polkrepko:weight=200
FreeSerif:style=Italic,ležeče:weight=80
FreeSans:style=Medium,navadno:weight=80
DejaVu Sans:style=Oblique:weight=80
FreeSans:style=Oblique,ležeče:weight=80
```

可以使用 `fc-list` 查询的重要参数包括：

表 13.2 *fc-list* 的参数

参数	含义和可能值
family	字体系列的名称，如 FreeSans。
foundry	字体的制造商，如 urw。
style	字型，如 Medium、Regular、Bold、Italic 或 Heavy。

参数	含义和可能值
lang	字体支持的语言，例如 de 表示德语，ja 表示日语，zh-TW 表示繁体中文，zh-CN 表示简体中文。
weight	字体粗细，例如 80 表示常规粗细，200 表示粗体。
slant	倾斜，通常 0 表示不倾斜，100 表示斜体。
file	包含字体的文件的名称。
outline	true 表示外框字体，false 表示其他字体。
scalable	true 表示可缩放字体，false 表示其他字体。
bitmap	true 表示位图字体，false 表示其他字体。
pixelsize	以像素为单位表示的字体大小。与 fc-list 一起使用时，此选项仅对位图字体有意义。

13.3 更多信息

安装包 xorg-x11-doc 和 howtoenh 以获得关于 X11 的详细信息。有关 X11 开发的更多信息，请参见该项目的主页：<http://www.x.org>。

手册页中详细描述了包 xorg-x11-driver-video 随附的多个驱动程序。例如，如果使用 radeon 驱动程序，在 man 4 radeon 中可找到有关此驱动程序的更多信息。

有关第三方驱动程序的信息位于 /usr/share/doc/packages/<package_name> 中。例如，x11-video-nvidiaG01 的文档在安装包之后位于 /usr/share/doc/packages/x11-video-nvidiaG01 中。

使用 FUSE 访问文件系统

FUSE 是用户空间中的文件系统 (*file system in userspace*) 的缩写。这表示您可以将文件系统作为非特权用户配置和装入。通常，此任务需要您是根用户。FUSE 自身是一个内核模块。它与插件组合，允许您扩展 FUSE 以访问几乎所有文件系统，如远程 SSH 连接、ISO 映像等

14.1 配置 FUSE

可以使用 FUSE 之前，必须安装包 `fuse`。根据要使用的文件系统，您需要不同包中的附加插件。使用 YaST 搜索这些包，并使用 `fuse` 或 `file system` 作为关键字。

一般您不必配置 FUSE，只需使用即可。但是建议创建一个组合所有安装点的目录。例如，可以创建目录 `~/mounts` 并在该处插入不同文件系统的子目录。

14.2 装入 NTFS 分区

NTFS（新技术文件系统，*New Technology File System*）是一些 Windows 版本（如 Windows NT、2000、XP 和 Vista）的默认文件系统。它取代了 FAT 文件系统。要将 Windows 分区作为普通用户装入，请如下操作：

- 1 转换为根用户并安装包 `ntfs-3g`。
- 2 创建目录 `/media/windows`。

3 确定您需要哪个 Windows 分区。使用 YaST 并启动分区程序模块以查看哪个分区属于 Windows，但不要更改任何内容。或者转换为根用户并执行 `/sbin/fdisk -l`。查找分区类型为 HPFS/NTFS 的分区。

4 以读写模式装入分区。使用相应的 Windows 分区替换占位符 *DEVICE*：

```
ntfs-3g /dev/DEVICE /media/windows
```

如果要以只读模式使用 Windows 分区，则追加 `-o`：

```
ntfs-3g /dev/DEVICE /media/windows -o ro
```

命令 `ntfs-3g` 使用当前用户 ID (`uid`) 和组 ID (`gid`) 装入给定设备。如果要为其他用户设置写权限，则使用命令 `id USER` 获取 `uid` 和 `gid` 值的输出。设置方式：

```
id tux
uid=1000(tux) gid=100(users) Gruppen=100(users),16(dialout),33(video)
ntfs-3g /dev/DEVICE /media/windows -o uid=1000,gid=100
```

在手册页中查找其他选项。

要卸载该资源，请使用：

```
fusermount -u /media/windows
```

14.3 使用 SSHFS 装入远程文件系统

SSH（安全外壳网络协议）可以用于通过安全通道在两台计算机之间交换数据。要通过 FUSE 建立 SSH 连接，请如下操作：

1 安装包 `sshfs`。

2 创建希望访问远程计算机的目录。建议使用 `~/mounts/HOST`。使用远程计算机的名称替换 *HOST*。

3 装入远程文件系统：

```
sshfs USER:HOST ~/mounts/HOST
```

使用相应的值替换 *USER* 和 *HOST*。

4 输入远程计算机的密码。

14.4 装入 ISO 文件系统

要查看 ISO 映像，您可以使用 `fuseiso` 包装入：

- 1 安装包 `fuseiso`。
- 2 创建目录 `~/mounts/iso`。
- 3 装入 ISO 映像：
`fuseiso ISO_IMAGE ~/mounts/iso`

仅可读取 ISO 映像的内容，不能重新写入。

14.5 可用 FUSE 插件

FUSE 依赖插件。下表列出常用插件。

表 14.1 可用 FUSE 插件

<code>fuseiso</code>	使用 CD-ROM 映像中的 ISO9660 文件系统装入映像
<code>ntfs-3g</code>	装入 NTFS 卷（有读写支持）
<code>sshfs</code>	基于 SSH 文件传输协议的文件系统客户端
<code>wdfs</code>	装入 WebDAV 文件系统

14.6 更多信息

请参见 FUSE 主页 <http://fuse.sourceforge.net> 以了解更多信息。

部分 III. 移动计算机

Linux 中的移动计算

移动计算主要与便携式计算机、PDA、手提电话以及它们之间的数据交换关联。移动硬件部件（如外部硬盘、闪存盘或数码相机）可连接到便携式计算机或台式机。移动计算方案中涉及了许多软件组件，一些应用程序是专门为移动定制的。

15.1 便携式计算机

便携式计算机的硬件不同于普通台式机的硬件。这是因为交换能力、占用空间和能耗能之类的标准都是要考虑的属性。移动硬件的制造商已开发了标准接口，如可用于扩展便携式计算机硬件的 PCMCIA（个人计算机内存卡国际协会）、迷你 PCI 和迷你 PCIe。该标准涉及内存卡、网络接口卡、ISDN 卡和调制解调器卡以及外部硬盘。

提示: SUSE Linux Enterprise Desktop 和 Tablet PC

SUSE Linux Enterprise Desktop 也支持 Tablet PC。Tablet PC 附带触摸板/数字转换器，使您可以使用数字笔甚至指尖代替鼠标和键盘直接在屏幕上编辑数据。它们的安装和配置与其他任何系统类似。有关 Tablet PC 安装和配置的介绍，请参见第 17 章 *使用 Tablet PC* [173]。

15.1.1 省电

由于在制造便携式计算机时加入了能量优化系统部件，这使得不必连接电源线即可使用便携式计算机。这些部件在省电方面所起的作用并不亚于操作系统。SUSE® Linux Enterprise Desktop 支持各种影响便携式计算机能耗的方法，在使用电池供电时，这些方法对计算机运行时间的影响各不相同。下面的列表按照省电方面作用从大到小排列：

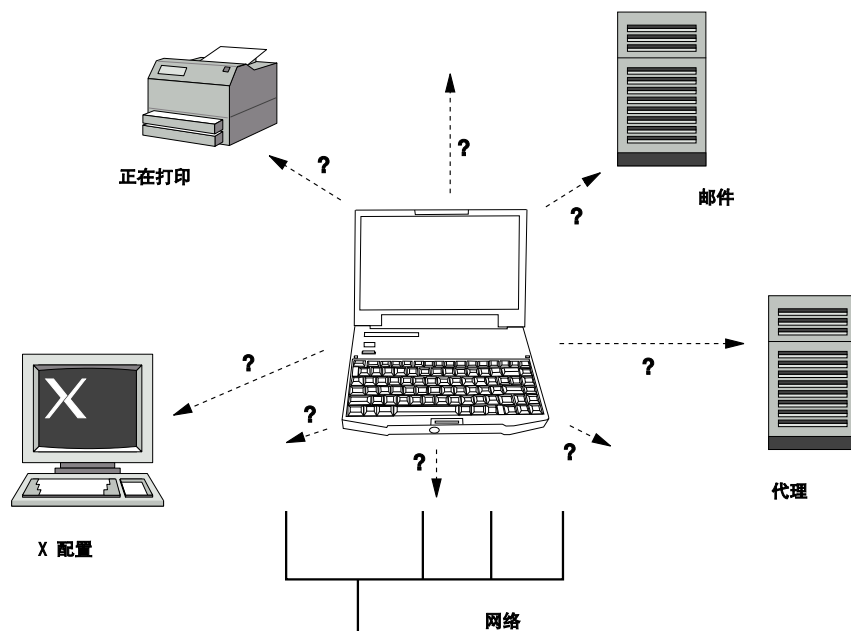
- 节制 CPU 流速.
- 在暂停期间关闭显示器.
- 手动调节显示器亮度.
- 断开不使用的支持热插拔的附件（USB CD-ROM、外部鼠标、不使用的 PCMCIA 卡、WLAN 等）。
- 在硬盘闲置时降低其转速.

有关特定于桌面的电源管理的更多信息，请参见第 1.15 节 “Controlling Your Desktop’s Power Management” (第 1 章 *Getting Started with the GNOME Desktop*, ↑*GNOME 用户指南*) 中 GNOME 电源管理器的使用方法。有关 KDE 电源管理小程序的更多信息可在第 8 章 *Controlling Your Desktop’s Power Management with KPowersave* (↑*KDE 用户指南*) 中获得。

15.1.2 在变化的操作环境中集成

在用于移动计算时，您的系统需要适应变化的操作环境。很多服务都依赖环境而且必须重配置底层客户端。SUSE Linux Enterprise Desktop 会为您处理该任务。

图 15.1 在现有环境中集成移动计算机



对于在小型家庭网络和办公网络之间往来通讯的便携式计算机，受影响的服务包括：

网络

这包括 IP 地址分配、域名解析、因特网连接以及与其他网络的连接。

打印

必须存在可用打印机的当前数据库和可用的打印服务器（具体取决于网络）。

电子邮件和代理

就像在打印中那样，当前必须存在一组相应的服务器。

X（图形环境）

如果您的便携式计算机暂时连接到投影机或外部监视器，则需要有其他显示配置。

SUSE Linux Enterprise Desktop 提供几种方法可将便携式计算机集成到现有的操作环境中：

NetworkManager

是为便携式计算机上的移动联网特别设计的。它能够简单而自动地在各种网络环境和网络类型（如无线 LAN 和以太网）之间切换。NetworkManager 支持无线局域网中的 WEP 和 WPA-PSK 加密。它也支持拨号连接（使用 `smpppd`）。这两种桌面环境（GNOME 和 KDE）均包含 NetworkManager 的前端。关于桌面小程序的更多信息，请参见第 22.4 节“使用 KDE NetworkManager 控件”[258]和第 22.5 节“使用 GNOME NetworkManager 小程序”[258]。

表 15.1 NetworkManager 的用例

我的电脑...	使用 NetworkManager
是便携式计算机	是
有时与不同网络连接	是
提供网络服务（例如 DNS 或 DHCP）	否
仅使用静态 IP 地址	否

在不应使用 NetworkManager 来处理网络配置时，请使用 YaST 工具配置联网。

SCPM

SCPM（系统配置配置文件管理）允许将系统的任意配置状态储存为一种称为“配置文件”的快照。可以为不同的情况创建配置文件。在变化的环境（家庭网络、办公网络）中操作系统时，这些配置文件十分有用。可以随时在配置文件间切换。要在系统上设置并运行 SCPM，请安装包 `kscpm`，并将配置文件选择器 KDE 小程序添加到面板中，使用“YaST 配置文件管理”模块启用 SCPM，并配置允许切换配置文件而无需输入 `root` 密码的用户。确定配置文件更改在系统重引导后是否生效还是在关闭后将丢弃这些更改。请确保所有资源组（即，例如网络和打印机服务）都为活动的。使用通过配置文

件选择器启动的 SUMF（SCPM 统一管理前端）工具继续创建实际配置文件。为所有希望使用该系统的不同安装程序创建配置文件。在配置文件间切换既可通过配置文件选择器小程序在运行的系统中进行，也可以通过 F3 键在系统引导时进行。切换配置文件时，SCPM 会自动将系统配置调整为所选配置文件中设置的新环境。

SLP

服务位置协议 (SLP) 简化了便携式计算机与现有网络的连接。没有 SLP，便携式计算机的管理员通常需要详细了解网络中可用的服务。使用 SLP 则可以向本地网络中的所有客户机广播某种服务是否可用。支持 SLP 的应用程序可以处理 SLP 发送的信息，并进行自动配置。SLP 甚至还可用于安装系统，而不必再费事地搜索适合的安装源。有关 SLP 的更多详细信息，请参见第 20 章 *网络中的 SLP 服务* [245]。

15.1.3 软件选择

在移动使用中有不同的特殊任务领域，它们由专用软件实现：系统监视（特别是电池充电）、数据同步和与外围设备及因特网的无线通讯。以下各节描述了 SUSE Linux Enterprise Desktop 为各项任务提供的最为重要的应用程序。

系统监视

SUSE Linux Enterprise Desktop 提供了两种 KDE 系统监视工具：

KPowersave

KPowersave 是可以在控制面板中显示充电电池状态的小程序。该图标将随电源类型调整显示。如果使用交流电，则显示一个小的插头图标。如果使用电池，则改为显示电池图标。提供 root 密码之后，相应的菜单会打开用于电源管理的 YaST 模块。这样就可以为不同电源设置系统的行为。

KSysguard

KSysguard 是一个独立的应用程序，它可以将所有可测量系统参数收集到一个监视环境中。KSysguard 提供用于 ACPI（电池状态）、CPU 负载、网络、分区和内存使用等方面的监视程序。它还可以查看和显示所有的系统进程。可以自定义收集到的数据的表示和过滤方式。可以监视不同数据页中的不同系统参数，也可以跨网络并行收集不同计算机上的数据。KSysguard 还可以在不具备 KDE 环境的计算机上作为守护程序运行。有关此程序的详细信息，请参见此程序中集成的帮助功能或 SUSE 帮助页。

在 GNOME 桌面中，请使用 GNOME 电源管理自选设置和系统监视程序。

同步数据

如果要在以下两种工作方式（在与网络断开的移动计算机上工作和在办公室中的联网工作站上工作）之间切换，则需要所有实例间保持同步处理数据。要同步的可能包括电子邮件文件夹、目录和单个文件，这些数据需要保持最新，以便在途中和办公室中处理。适用于这两种情况的解决方案如下：

同步电子邮件

在办公室网络中使用 IMAP 帐户储存电子邮件。随后可以从工作站使用任意断开连接但支持 IMAP 的电子邮件客户端（如 *GNOME 用户指南* (*↑GNOME 用户指南*) 和 *KDE 用户指南* (*↑KDE 用户指南*) 中所述的 Mozilla Thunderbird Mail、Evolution 或 KMail）来访问这些电子邮件。必须对电子邮件客户机进行配置，以便始终从同一文件夹访问已发送邮件。这样能确保在完成同步过程之后可以提供所有信件及其状态信息。使用邮件客户机中实施的 SMTP 服务器来发送邮件，取代系统范围内使用的 MTA postfix 或 sendmail 来接收有关未发送邮件的可靠反馈。

同步文件和目录

有多个实用程序适合在便携式计算机和工作站之间同步数据。

无线通讯

便携式计算机不仅可以通过缆线连接家庭或办公网络，而且可以无线连接到其他计算机、外设、手提电话或 PDA。Linux 支持三种类型的无线通讯：

WLAN

WLAN 在这些无线技术中覆盖范围最广，是唯一一种适用于大型（有时甚至是在空间上分离的）网络的操作技术。单独的计算机可以通过互连形成独立的无线网络或访问因特网。称为访问点的设备充当支持 WLAN 设备的基站，并作为访问因特网的中介。移动用户可以在多个访问点之间切换，这取决于所在位置以及哪个访问点提供的连接最佳。类似移动电话的情况，WLAN 用户可以访问一个大型网络，而不必被集中到某个位置来访问这个网络。有关 WLAN 的详细信息请参见第 19.1 节“无线 LAN” [235]。

蓝牙

蓝牙技术是所有无线技术中应用范围最广的技术。与 IrDA 一样，蓝牙技术可用于计算机（便携式计算机）和 PDA 或手提电话之间的通信。它还可用于

于连接视线范围内的多台计算机。蓝牙技术还可用于连接键盘或鼠标之类的无线系统部件。但这种技术的覆盖范围还不够大，无法将远程系统连接到网络中。WLAN 是穿越墙壁之类的有形障碍物进行通讯的首选技术。

IrDA

IrDA 是覆盖范围最小的无线技术。通讯双方必须在彼此的视线范围之内。无法穿越墙壁这样的障碍物。将文件从便携式计算机传送到手提电话就是 IrDA 的一种应用方式。使用 IrDA 即可覆盖由便携式计算机到手提电话之间的较短路径。要在较大范围内将文件传输给接收方，则需要通过移动网络来处理。IrDA 的另一种应用方式就是在办公室中无线传送打印作业。

15.1.4 数据安全

要防止他人未经授权访问您的便携式计算机上的数据，您最好同时采用多种方式。可以在以下方面采取各种可能的安全措施：

防止被盗

始终尽可能地利用实物来防止您的系统被盗。零售店中就出售各种防盗工具，如锁链。

强大的身份验证

除了通过登录名和密码的标准身份验证外，还使用生物特征身份验证。SUSE Linux Enterprise Desktop 支持指纹身份验证。有关细节，请参见第 3 章 *Using the Fingerprint Reader* (↑安全指南)。

保护系统中的数据

重要数据不仅要在传送过程中加密，而且要在硬盘上加密。这样即使被盗也能保证数据不外泄。第 12 章 *Encrypting Partitions and Files* (↑安全指南) 中对如何使用 SUSE Linux Enterprise Desktop 创建加密分区进行了描述。在使用 YaST 添加用户时还可以创建加密的用户主目录。

重要：数据安全性与挂起磁盘

在发生挂起磁盘事件期间，不会卸载加密的分区。因此，任何人只需窃取硬件然后对硬盘发出 `resume` 命令就可以获取这些分区上的所有数据。

网络安全

无论采用哪种形式传送数据，任何形式的传送都应受到保护。有关 Linux 和网络的常见安全性问题，请参见第 1 章 *Security and Confidentiality* (↑安全指南)。有关无线联网的安全措施，请参见 第 19 章 *无线通讯* [235]。

15.2 移动硬件

SUSE Linux Enterprise Desktop 支持通过防火墙 (IEEE 1394) 或 USB 自动检测移动储存设备。术语 *移动储存设备* 适用于任何种类的防火墙或 usb 硬盘、USB 闪存驱动器，或数码相机。这些设备在经相应的接口和系统连接之后，将立刻被检测到并配置。GNOME 和 KDE 的文件管理器均可灵活操作移动硬件项目。要安全卸载这些媒体的任何一项，请使用任意文件管理器的安全删除 (KDE) 或卸载卷 (GNOME) 功能。在 *GNOME 用户指南* (↑*GNOME 用户指南*) 和 *KDE 用户指南* (↑*KDE 用户指南*) 中更加详细地描述了桌面对于可卸媒体的处理。

外部硬盘 (USB 和火线)

一旦系统正确识别到外部硬盘，其图标将显示在文件管理器中。单击该图标将显示该驱动器的内容。可以在此创建文件夹和文件，并执行编辑或删除操作。要将系统指定的硬盘名称重命名，请右击该图标，从打开的菜单中选择相应的菜单项。只有在文件管理器中才能显示这种名称更改。将设备装入 /media 中的描述符将不受影响。

USB 闪存盘

系统会按照处理外部硬盘的方式来处理这些设备。同样也可以重命名文件系统中的项。

数码相机 (USB 和火线)

系统识别出的数码相机也作为外部驱动器显示在文件管理器的概览中。KDE 允许读取和访问 URL `camera:/` 中的图片。随后可以使用 digiKam 或 f-spot 对图像进行处理。用 GIMP 进行照片的高级处理。关于 digiKam、f-spot 和 GIMP 的简短介绍，请参见第 24 章 *Managing Your Digital Image Collection* (↑应用程序指南)、第 25 章 *Managing Your Digital Image Collection with F-Spot* (↑应用程序指南) 和第 23 章 *Manipulating Graphics with The GIMP* (↑应用程序指南)。

15.3 手提电话和 PDA

台式计算机系统或便携式计算机可以通过蓝牙或 IrDA 与手提电话进行通信。有些手提电话型号两种协议都支持，另一些则只支持其中的一种。这两种协议的使用范围以及相应的展开文档都已在“[无线通讯](#)”一节 [158] 中描述。手提电话自带的手册中对如何在手提电话上配置这些协议进行了描述。

Evolution 和 Kontact 中已内置与 Palm, Inc. 制造的手持设备进行同步的支持功能。最初与手持设备连接时，无论使用哪种应用程序都可以借助向导轻松连接。一旦配置了针对 Palm Pilots 的支持，则需要确定应该同步哪种数据（地址、约会等）。有关详细信息，请参见 *GNOME 用户指南* ([↑GNOME 用户指南](#)) 和 *KDE 用户指南* ([↑KDE 用户指南](#))。

程序 opensync 提供了更高级的同步解决方案（请参见包 libopensync、msynctool 及不同设备各自的插件）。

15.4 更多信息

<http://tuxmobil.org/> 是与移动设备和 Linux 有关的所有问题的集中参考来源。该 Web 站点的各个章节论述了便携式计算机、PDA、手提电话和其他移动硬件的软硬件问题。

<http://www.linux-on-laptops.com/> 中也提供了与 <http://tuxmobil.org/> 类似的参考资源。可以在此站点中找到有关便携式计算机和手持设备的信息。

SUSE 维护着一个德文邮件列表，专门讨论便携式计算机这一主题。请参见 <http://lists.opensuse.org/opensuse-mobile-de/>。在该列表中，用户和开发人员讨论了有关 SUSE Linux Enterprise Desktop 中移动计算的各方面问题。用英文发送的邮件都有答复，但存档信息中大部分都只有德文信息。请使用 <http://lists.opensuse.org/opensuse-mobile/> 用英文发送邮件。

关于 OpenSync 的信息可以在 <http://en.opensuse.org/OpenSync> 上找到。

电源管理

电源管理对于便携式计算机特别重要，但对于其他系统也是有用的。所有现代计算机（便携式计算机、台式机和服务器）上都提供有 ACPI（高级配置与电源接口）。电源管理技术需要合适的硬件和 BIOS 例程。大多数便携式计算机、许多目前的台式机和服务器都符合这些要求。还可以通过控制 CPU 频率调节以达到省电或降低噪音的目的。

16.1 省电功能

省电功能不仅对便携式计算机的移动使用很重要，而且对台式机系统也很重要。ACPI 中的主要功能和它们的用法为：

待机

不支持.

暂挂（到内存）

此方式将整个系统状态写入 RAM。随后，除 RAM 外，整个系统都进入休眠状态。在此状态下，计算机消耗的电量非常少。此状态的优点是无需引导和重新启动应用程序就可以在数秒内将工作恢复到原来的进度。此功能对应于 ACPI 状态 S3。对此状态的支持仍在开发中，因此目前主要依靠硬件来实现支持。

休眠（暂挂到磁盘）

在此运行方式下，将整个系统状态写入硬盘并关闭系统电源。至少要有一个像 RAM 一样大的交换分区才能写入所有活动的数据。从该状态重激活大约需要 30 至 90 秒的时间。将恢复到暂停之前的状态。某些制造商提供这

种方式的有用的混合变体（例如 IBM Thinkpad 中的 RediSafe）。对应的 ACPI 状态是 S4。在 Linux 中，由独立于 ACPI 的内核例程执行暂挂到磁盘。

电池监视

ACPI 检查电池充电状态并提供相关信息。另外，当达到临界电量状态时，它将协调要执行的操作。

自动关闭电源

关闭后，将关闭计算机的电源。当在电池电量用完前立即执行自动关闭时，此功能特别重要。

处理器速度控制

在 CPU 方面，有三种方法可以节省电能：频率和电压调节（也称为 PowerNow! 或 Speedstep）、节流和让处理器休眠（C 状态）。根据计算机的运行方式，还可以将这三种方法结合起来使用。

16.2 ACPI

ACPI（高级配置和电源接口）支持操作系统设置和控制各个硬件部件。ACPI 可以取代 PnP 和 APM。它提供有关电池、AC 适配器、温度、风扇和系统事件（例如“合上机盖”或“电池电量低”）的信息。

BIOS 提供包含有关各个部件和硬件访问方法信息的表。操作系统使用这些信息执行指派中断或激活和取消激活部件等任务。因为操作系统执行 BIOS 中储存的命令，所以功能取决于 BIOS 实施。/var/log/boot.msg 中报告了 ACPI 能够检测并装载的表。有关对 ACPI 问题进行故障诊断的详细信息，请参见第 16.2.3 节“故障诊断” [166]。

16.2.1 控制 CPU 性能

CPU 可以采用三种省电方法。根据计算机的运行方式，还可以将这三种方法结合起来使用。省电还意味着系统温度不会升得过高并且激活风扇的频率会降低。

频率和电压调节

PowerNow! 和 Speedstep 是 AMD 和 Intel 为这一技术指定的名称。但是，其他制造商的处理器中也应用了这一技术。CPU 的时钟频率及其核心电压同时降低，因而采用这一技术所节省的电量远远超过了线性省电量。这意味

着，如果频率减半（一半的性能），所节省的电量远不止一半。此技术独立于ACPI。可使用两种主要的方法来执行CPU频率调节：通过内核本身或通过用户空间应用程序。因此，可以在 `/sys/devices/system/cpu/cpu*/cpufreq/` 下设置不同的内核管理器。

userspace governor

如果设置了用户空间管理器，则内核会将对CPU频率调节的控制指定给用户空间应用程序（通常是守护程序）。在SUSE Linux Enterprise Desktop分发中，此守护程序是 `powersaved` 程序包。使用此实施时，将根据当前系统负载调整CPU的频率。默认情况下，将使用某个内核实施。但是，在某个硬件上或对于特定处理器或驱动器，用户空间实施仍是唯一的工作解决方法。

ondemand governor

它是动态CPU频率策略的内核实施，应该可在大多数系统上运行。只要系统负载过高，CPU频率将立即增加。它在系统负载较低时也较低。

conservative governor

此管理器与按需实现相似，只是使用更保守的策略。对于CPU频率增加之前的特定时间内，系统的负载必须很高。

powersave governor

静态地将cpu频率设置为最低。

performance governor

静态地将cpu频率设置为最高。

节流时钟频率

此技术将忽略一定百分比的CPU时钟信号脉冲。如果节流25%，则将忽略四分之一的脉冲，如果节流87.5%，则只有八分之一的脉冲到达处理器。但是，采用这种方法所节省的电量稍微低于线性省电量。通常，只有在频率调节不可用或要最大程度节省电量时才使用节流。此技术也必须由特殊的进程控制。系统接口是 `/proc/acpi/processor/*/throttling`。

使处理器进入休眠状态

操作系统在处理器不执行任何任务时使处理器进入休眠状态。在这种情况下，操作系统向CPU发送一个 `halt` 命令。有三种状态：C1、C2和C3。最经济的状态是C3，在这种状态下，连处理器高速缓存与主存之间的同步都将暂停。因此，只有在没有任何其他设备通过总线主控芯片活动修改主

储存器的内容时才能应用此状态。某些驱动程序禁止使用 C3。当前状态显示在 `/proc/acpi/processor/*/power` 中。

只有当处理器忙时，才需要进行频率调节和节流，这是因为当处理器处于空闲状态时总是会应用最经济的 C 状态。如果 CPU 忙，则建议采用的省电方法是频率调节。处理器经常只在部分负载的状态下工作。在这种情况下，可以以较低的频率运行。通常，由内核按需管理器 (kernel on demand governor) 或一个守护程序（如 powersaved）控制的动态频率调节是最佳方法。如果使用电池工作或如果您想让计算机冷却或安静，则静态设置为低频率会非常有用。

节流应作为最后没有办法时采用的方法，例如，虽然系统负载很高，但为延长电池工作时间而采用节流。但是，如果节流程度过高，某些系统将不会正常运行。此外，如果 CPU 处理的任务量很少，则 CPU 节流就没什么作用。

16.2.2 ACPI 工具

一系列相对全面的 ACPI 实用程序包含这样一些工具：只显示信息（例如，电池电量水平和温度）的工具（`acpi`、`klaptopdaemon` 和等）、简化对 `/proc/acpi` 中的结构进行访问的工具或协助监视更改的工具（`akpi`、`acpiw` 和 `gtkacpiw`）以及用于编辑 BIOS 中 ACPI 表的工具（包 `pmtools`）。

16.2.3 故障诊断

问题有两种不同的类型。一种是内核的 ACPI 代码可能包含未及时检测出的错误。在这种情况下，可以通过下载获得解决方案。而另一种更常见的问题，是由 BIOS 引起的。有时，会故意将一些不符合 ACPI 规范的配置集成在 BIOS 中，用于避免其他常用操作系统中 ACPI 实施中的错误。在 ACPI 实施中有严重错误的硬件部件会被记录在一个黑名单中，防止 Linux 内核对这些部件使用 ACPI。

在遇到问题时，首先要做的是更新 BIOS。如果计算机根本未引导，则使用以下引导参数之一可能会解决问题：

`pci=noacpi`

不使用 ACPI 配置 PCI 设备。

`acpi=ht`

只执行简单的资源配置。不要将 ACPI 用于其他目的。

acpi=off
禁用 ACPI。

警告: 不使用 ACPI 引导会出现问题

某些较新的计算机（特别是 SMP 系统和 AMD64 系统）需要 ACPI 以正确配置硬件。在这些计算机上，禁用 ACPI 可能会产生问题。

有时，计算机会对通过 USB 或 FireWire 挂接的硬件感到困惑。如果一台计算机拒绝引导，请拔下所有不需要的硬件，然后再次重试。

引导后，用命令 `dmesg | grep -2i acpi` 来监视系统的引导消息（或所有消息，因为问题可能不是由 ACPI 引起的）。如果在分析 ACPI 表时出错，则最重要的表 (DSDT) 可替换为更高的版本。在这种情况下，将忽略 BIOS 中有问题的 DSDT。中对这一过程进行了介绍。第 16.4 节“查错”[169]

在内核配置中，可以使用开关来激活 ACPI 调试消息。如果已编译并安装了具有 ACPI 调试功能的内核，则支持对详细信息执行错误专家搜索。

如果遇到 BIOS 或硬件问题，则最好与制造商联系。特别是如果制造商不常对 Linux 提供支持，他们就应该面对这些问题。只有在制造商意识到有很多客户在使用 Linux 时，他们才会重视这一问题。

更多信息

- <http://www.cpqlinux.com/acpi-howto.html>（详细的 ACPIHOWTO 文档，包含 DSDT 增补程序）
- <http://www.intel.com/technology/iapc/acpi/index.htm>（高级配置和电源接口）
- <http://www.lesswatts.org/projects/acpi/>（Sourceforge 中的 ACPI4Linux 项目）
- <http://www.poupinou.org/acpi/>（Bruno Ducrot 开发的 DSDT 增补程序）

16.3 硬盘的休眠

在 Linux 中，如果不使用硬盘，则可以使硬盘完全进入休眠状态，或者在更经济或更安静的方式下运行。在目前的便携式计算机上，您无需手动关闭硬盘，因为硬盘会在不运行时自动进入经济的运行方式。但是，如果要最大限度地省电，请尝试使用以下一些方法。

`hdparm` 应用程序可用于修改多种硬盘设置。选项 `-y` 将硬盘立即切换到待机方式。`-Y` 使硬盘进入休眠状态。`hdparm -S x` 会使硬盘在一段时间（未活动）后减慢运行速度。将 `x` 替换如下：0 表示禁用此机制，导致硬盘持续运行。值 1 到 240 表示的时间为所选的值乘以 5 秒。值 241 到 251 对应的时间分别是 30 分钟的 1 到 11 倍。

使用选项 `-B` 可以控制硬盘的内部省电选项。在 0 到 255 之间选择一个值，0 表示最大省电方式，255 表示最大吞吐量方式。结果取决于所使用的硬盘，难以估算。要让硬盘安静一些，请使用选项 `-M`。在 128 到 254 之间选择一个值，128 表示最安静，254 表示速度最快。

通常，让硬盘进入休眠状态并不容易。在 Linux 中，大量的进程对硬盘执行写操作，因而会经常将其唤醒。因此，一定要了解 Linux 如何处理需要写入硬盘的数据。首先，在 RAM 中对所有数据进行缓冲。缓冲区正在由 `pdflush` 守护程序监视。当数据达到一定的有效期限限制或缓冲区已被填充到一定程度时，就会清理缓冲区，将其中的内容写入硬盘。缓冲区大小是动态的，取决于内存的大小和系统负载。默认情况下，将 `pdflush` 设置为较短的时间间隔可以获得最好的数据完整性。它会每 5 秒钟检查一次缓冲区并将数据写入硬盘。以下变量很有用：

```
/proc/sys/vm/dirty_writeback_centisecs
```

包含直到 `pdflush` 线程唤醒的延迟（以百分之一秒为单位）。

```
/proc/sys/vm/dirty_expire_centisecs
```

定义最晚在什么时间范围之后应写出未写入页。默认值是 3000，表示 30 秒。

```
/proc/sys/vm/dirty_background_ratio
```

`pdflush` 开始写入未写入页之前未写入页的最大百分比。默认值是 5%。

`/proc/sys/vm/dirty_ratio`

当未写入页超出总内存的此百分比后，将强制进程在其时间范围内写入未写入缓冲区，而不是继续写入。

警告: 对数据完整性的损害

更改 `pdflush` 守护程序设置将损害数据完整性。

除了这些进程之外，日记文件系统（例如 `ReiserFS` 和 `Ext3`）独立于 `pdflush` 写入它们的元数据，这也会妨碍硬盘减慢运行速度。为了避免这种情况，已为移动设备开发了特殊的内核扩展。有关详细信息，请参见 `/usr/src/linux/Documentation/laptop-mode.txt`。

另一个重要因素是活动程序的行为方式。例如，好的编辑器会定期将当前已修改文件的隐藏备份写入硬盘，而这会唤醒磁盘。可以禁用此类功能，但这会影响数据的完整性。

在此连接中，邮件守护程序 `postfix` 使用变量 `POSTFIX_LAPTOP`。如果将此变量设为 `yes`，则 `postfix` 访问硬盘的频率将显著降低。

在 `SUSE Linux Enterprise Desktop` 中，这些技术受 `laptop-mode-tools` 控制。

16.4 查错

文件 `/var/log/messages` 中记录了所有错误消息和警报。如果您未能找到所需的信息，请使用文件 `/etc/sysconfig/powersave/common` 中的 `DEBUG` 增加 `powersave` 消息的详细程度。请将变量的值增加到 7，或甚至增加到 15，然后重新启动守护程序。`/var/log/messages` 中更详细的错误消息应有助于您找到错误。以下章节讲述省电模式和不同休眠模式中最常见的问题。

16.4.1 硬件支持已激活 ACPI，但功能不工作

如果使用 `ACPI` 时遇到问题，请使用命令 `dmesg|grep -i acpi` 在 `dmesg` 的输出中搜索 `ACPI` 特定的消息。可能需要更新 `BIOS` 来解决问题。请转到便携式计算机制造商的主页，查找已更新的 `BIOS` 版本，然后安装它。要求制造商

遵循最新的 ACPI 规范。如果在更新 BIOS 后错误仍然存在，则按以下步骤用已更新的 DSDT 替换 BIOS 中有问题的 DSDT 表。

- 1 从 <http://acpi.sourceforge.net/dsdt/index.php> 为您的系统下载 DSDT。检查是否已解压缩并编译了此文件，如果文件扩展名是 .aml（ACPI 计算机语言），则表明已完成这些操作。如果是这种情况，请继续执行第 3 步。
- 2 如果下载的表的文件扩展名是 .asl（ACPI 源语言），则必须使用 iasl（pmtools 包）对其进行编译。为此，请输入命令 `iasl -sa file.asl`。
- 3 将文件 DSDT.aml 复制到任何位置（建议的位置为 /etc/DSDT.aml）。编辑 /etc/sysconfig/kernel 并相应地调整指向 DSDT 文件的路径。启动 mkinitrd（包 mkinitrd）。一旦安装了内核并使用 mkinitrd 创建了 initrd，引导系统时就会集成并装载已修改的 DSDT。

16.4.2 CPU 频率不工作

请参考内核源代码 (kernel-source) 查看是否支持您的处理器。您可能需要特殊内核模块或模块选项来激活 CPU 频率控制。/usr/src/linux/Documentation/cpu-freq/* 中提供了此信息。

16.4.3 暂挂和待机不工作

ACPI 系统由于 DSDT 实现 (BIOS) 有问题，可能在暂挂和待机中会遇到问题。如果出现这种情况，请更新 BIOS。

当系统尝试卸载有问题的模块时，会停止系统或不触发暂挂事件。如果您未卸载模块或停止阻止成功暂停的服务，也会发生相同的情况。在这两种情况下，尝试确定阻止采用休眠方式的有问题的模块。日志文件 /var/log/pm-suspend.log 包含关于所发生的情况以及哪里可能有错误的详细信息。修改 /usr/lib/pm-utils/defaults 中的 SUSPEND_MODULES 变量以在暂挂或待机之前卸载有问题的模块。

请参见 <http://www.opensuse.org/Pm-utils> 和 <http://www.opensuse.org/S2ram> 以获取有关如何修改暂挂和继续进程的更多详细信息。

16.5 更多信息

- <http://www.opensuse.org/S2ram> — 如何使“暂挂到 RAM”工作
- <http://www.opensuse.org/Pm-utils> — 如何修改常规暂挂框架

使用 Tablet PC

SUSE® Linux Enterprise Desktop 自带对 Tablet PC 的支持。从下面可了解如何安装和配置 Tablet PC 并发现接受数字笔输入的一些有用的 Linux* 应用程序。

支持以下 Tablet PC:

- 具有序列 Wacom 设备的 Tablet PC，例如 ACER TM C30x 系列、Fujitsu Lifebook T 系列 (T30xx/T40xx/T50xx)、Gateway C-140X/E-295C、HP Compaq TC1100/TC4200/TC4400、2710p/2730p、IBM/Lenovo X41t/X61t、LG LT20、Motion M1200/M1400、OQO 02、Panasonic Toughbook CF-18、Toshiba Portege/Tecra M 系列、Satellite R15/R20。
- 具有 Wacom USB 设备的 Tablet PC，例如 ASUS R1E/R1F、Gateway C-120X/E-155C、HP Pavilion tx2000/tx2100/tx2500 系列。
- 具有 FinePoint 设备的 Tablet PC，例如 Gateway C210X/M280E/CX2724、HP Compaq TC1000。
- 具有触摸屏设备的 Tablet PC，例如 Asus R2H、Clevo TN120R、Fujitsu Siemens Computers P-Series、LG C1、Samsung Q1/Q1-Ultra。

安装 Tablet PC 包并正确配置数字转换器之后，用手写笔（也称手写输入笔，可以用于以下操作及应用程序）输入：

- 登录到 KDM 或 GDM
- 在 KDE 和 GNOME 桌面上解除屏幕锁定
- 也可以由其他定点设备（如鼠标或触摸板）触发的操作，例如，在屏幕上移动光标、启动应用程序、关闭窗口、调整窗口大小、移动窗口、移动窗口焦点及拖放对象
- 在 X Window System 中使用手势识别
- 用 GIMP 绘图
- 用如 Jarnal 或 Xournal 之类的应用程序记录或绘制或者用 Dasher 编辑大量文本。

注意：安装时需要键盘或鼠标

安装 SUSE Linux Enterprise Desktop 期间，手写笔无法用作输入设备。如果您的 Tablet PC 中没有装备内置键盘或触摸板，请将外部键盘或鼠标连接到 Tablet PC 上供安装系统时使用。

17.1 安装 Tablet PC 包

Tablet PC 所需的包包含在 TabletPC 安装模式中 — 如果已在安装期间选择，应已在系统上安装以下包：

- `cellwriter`：基于字符的手写输入板
- `jarnal`：基于 Java 的记录应用程序
- `wacom-kmp(-default)`：具有 USB Wacom 设备的 Tablet PC 的内核驱动程序
- `xournal`：用于记录和绘制的应用程序
- `xstroke`：X Window System 的手势识别程序

- `xvkbd`: X Window System 的虚拟键盘
- `x11-input-fujitsu`: Fujitsu P 系列 手写板的 X 输入模块
- `x11-input-evtouch`: 一些具有触摸屏的 Tablet PC 的 X 输入模块
- `x11-input-wacom`: Wacom 手写板的 X 输入模块
- `x11-input-wacom-tools`: 为 Wacom 手写板进行配置、诊断并建库

如果未安装这些包，请通过命令行手动安装所需包，或在 YaST 中对安装选择 TabletPC 模式。

17.2 配置手写板设备

您可以在安装过程中在硬件配置屏幕通过更改图形卡选项配置 Tablet PC（不包含具有触摸屏的 Tablet PC）。或者可以在安装后任何时间配置（内部或外部）手写板设备。

- 1 通过命令行或通过按 **Alt + F2** 然后输入 `sax2` 来启动 SaX2。
- 2 如果使用 Wacom 或 Finepoint 设备，请单击手写板显示手写板属性。

如果使用具有触摸屏的 Tablet PC，请单击触摸屏。

- 3 从右侧列表选择 *Tablet PC* 作为供应商并选择手写板的名称，然后选中激活此手写板。

如果您的计算机未列出，而您确信具有 Wacom 设备，请选择 *Wacom ISDV4 Tablet PC (SERIAL)* 或 *Wacom ISDV4 Tablet PC (USB)*。

- 4 切换到电子笔选项卡并确保以下选项已激活：添加笔和添加橡皮。如果您有具有触摸屏的 Tablet PC，还要激活添加触板。

- 5 单击确定保存更改。

完成 X Window 系统配置之后，通过注销重新启动 X 服务器。也可以离开用户界面并在虚拟控制台中运行 `init 3 && init 5`。

配置手写板设备之后，现在即可以使用笔（或根据您的 Tablet PC 用手指）作为输入设备。

17.3 使用虚拟键盘

要登录 KDE 或 GNOME 桌面或解除屏幕锁定，您可以照常输入用户名和密码或通过登录字段下方显示的虚拟键盘 `xvkbd` 输入。要配置键盘或访问集成帮助，请单击左下角的 `xvkbd` 字段打开 `xvkbd` 主菜单。

如果输入不可见（或未传输到所需的窗口），请通过单击 `xvkbd` 中的焦点然后单击应获取键盘事件的窗口而重定向焦点。

图 17.1 `xvkbd` 虚拟键盘



如果想在登录后使用 `xvkbd`，请从主菜单将其启动或从外壳输入 `xvkbd`。

17.4 旋转显示器

使用 `KRandRTray` (KDE) 或 `gnome-display-properties` (GNOME) 直接手动旋转显示内容或调节显示内容大小。`KRandRTray` 和 `gnome-display-properties` 都是针对 X 服务器的 `RANDR` 扩展的小程序。

从主菜单启动 `KRandRTray` 或 `gnome-display-properties`，或者输入 `krandrtray` 或 `gnome-display-properties` 从壳层启动这个小程序。启动相应小程序之后，小程序图标通常将添加到系统盘。如果 `gnome-display-properties` 图标在系统盘中未自动显示，请务必在监视器分辨率设置对话框中激活在面板中显示显示器。

要使用 KRandRTray 旋转显示器，请右键单击图标并选择配置显示器。从配置对话框选择所需方向。

要使用 gnome-display-properties 旋转显示器，请右键单击图标并选择所需方向。显示器将立即倾斜到新的方向。同时，图形手写板的方向也发生改变，因此它仍然可以正确解释手写笔的活动。

如果更改桌面方向时遇到问题，请参见第 17.7 节“查错”[181]以获取更多信息。

17.5 使用手势识别

SUSE Linux Enterprise Desktop 同时包含 CellWriter 和 xstroke 以识别手势。两个应用程序都接受使用笔或其他指向设备执行的手势作为 X Window 系统上应用程序的输入。

17.5.1 使用 CellWriter

通过 CellWriter，您可以将字符写入单元格网格中，写的内容会便于按字符识别。写完之后，您可以将输入发送到当前有焦点的应用程序。可以使用 CellWriter 识别手势之前，需要培训应用程序使其识别您的手写内容。您需要培训某个键映射的每个字符（未培训的字符不会激活，从而不能使用）。

过程 17.1 培训 CellWriter

- 1 从主菜单中或从命令行使用 `cellwriter` 启动 CellWriter。在第一次启动时，CellWriter 会自动以培训模式启动。在培训模式中，它会显示当前所选键映射的一组字符。
- 2 将要用于一个字符的手势输入相应字符的单元格。第一次输入时，背景颜色更改为白色，而字符本身会以浅灰色显示。重复该手势多次，直到该字符颜色更改为黑色。未培训的字符会在浅灰色或棕色背景中显示（因桌面颜色主题而异）。
- 3 重复此步骤，直到为 CellWriter 培训了所需的所有字符。
- 4 如果要培训 CellWriter 输入另一种语言，请单击设置按钮并从语言选项卡选择语言。关闭配置对话框。单击培训按钮并从 CellWriter 窗口右下角的下拉框中选择键映射。现在为新的键映射重复培训。

5 完成键映射的培训之后，单击 *培训* 按钮切换到普通模式。

在普通模式中，CellWriter 窗口显示可用于输入手势的两个空单元格。单击 *Enter* 按钮后这些字符才会发送到另一个应用程序，因此可以在使用字符作为输入之前更正或删除它们。已识别为低可信度的字符将高亮显示。要更正输入，请使用在右键单击单元格显示的上下文菜单。要删除字符，请使用笔的橡皮或按鼠标中键清除单元格。在 CellWriter 中完成输入之后，通过单击应用程序的窗口定义哪个应用程序应收到输入。然后通过单击 *Enter* 将输入发送到该应用程序。

图 17.2 使用 CellWriter 识别手势



如果在 CellWriter 中单击 *键* 按钮，将会得到一个虚拟键盘，可以使用其代替手写识别。

要隐藏 CellWriter，请关闭 CellWriter 窗口。现在该应用程序显示为系统盘中的图标。要再次显示输入窗口，请单击系统盘中的图标。

17.5.2 使用 Xstroke

使用 *xstroke*，在 X Window 系统中，可以将手势用于手写笔或其他定点设备，以此作为应用程序输入。*xstroke* 字母表是类似于 Graffiti* 字母表的 *unistroke* 字母表。一旦将其激活，它会将输入发送至当前聚焦的窗口中。

- 1 从主菜单启动 *xstroke*，或从外壳输入 *xstroke*。这将在您的系统盘中添加铅笔图标。
- 2 启动要使用手写笔创建文本的应用程序（例如，终端窗口、文本编辑器或 OpenOffice.org 记录器）。
- 3 要激活手势识别方式，请单击一次铅笔图标。

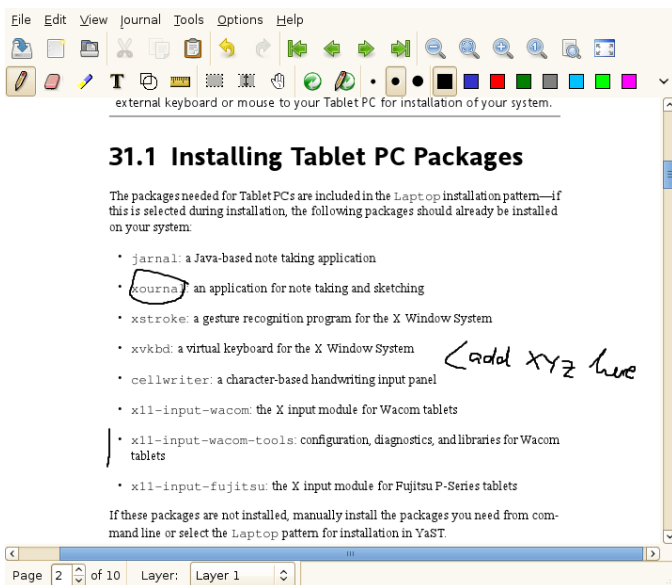
- 4 用手写笔或其他定点设备在图形手写板上执行手势。`xstroke` 截获手势并将它们传送到聚焦的应用程序窗口中显示的文本中。
- 5 要在不同窗口中切换聚焦，请用手写笔单击所需窗口并在按住鼠标一会儿（或使用桌面控制中心定义的键盘快捷方式）。
- 6 要停用手势识别方式，请再次单击铅笔图标。

17.6 用手写笔记录 and 绘制

要用手写笔创建绘图，可以使用如 GIMP 之类的专业图形编辑器或者尝试 Xournal 或 Jarnal 之中的一个记录应用程序。用 Xournal 或 Jarnal，您可以用手写笔记、创建绘图或为 PDF 文件添加注释。作为一个适用于多个平台的基于 Java 的应用程序，Jarnal 也提供基本协作功能。有关详细信息，请参见 <http://www.dklevine.com/general/software/tcl000/jarnal-net.htm>。保存内容时，Jarnal 以存档格式 (*.jaj) 储存数据，这其中也包含 SVG 格式的文件。

从主菜单启动 Jarnal 或 Xournal 或者从外壳输入 Jarnal 或 Xournal。例如，要给 Xournal 中的 PDF 文件添加注释，请选择文件 > 注解 *PDF* 然后在文件系统中打开 PDF 文件。使用手写笔或其他定点设备注解 PDF，然后通过文件 > 打印到 *PDF* 来保存更改。

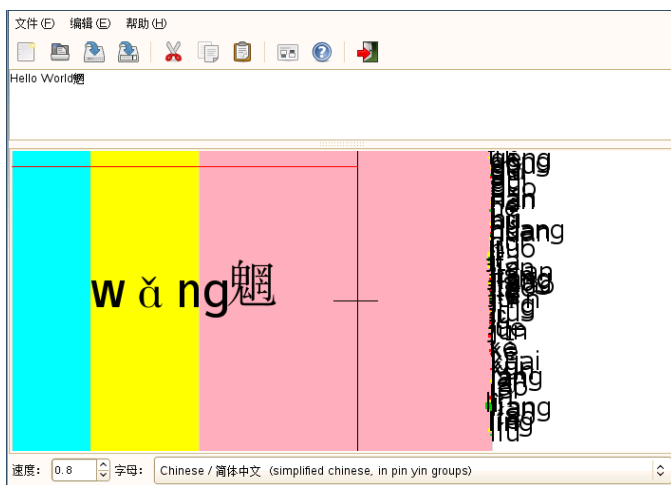
图 17.3 用 Xournal 注解 PDF



Dasher 是另一个有用的应用程序。它适用于键盘输入不实际或不可用的情况。只需稍加练习，您就可以只需使用手写笔（或其他输入设备——甚至装备了眼球跟踪器）即能熟练输入大量文本。

从主菜单启动 Dasher 或从外壳输入 dasher。沿一个方向移动手写笔，应用程序将开始放大到右侧的字母区。从穿过中间十字准线的字母开始，创建或预测文本，且将文本打印到窗口的上方。要停止或开始写入，请用手写笔单击一下显示器。在窗口底部修改缩放速度。

图 17.4 用 Dasher 编辑文本



Dasher 概念适用于多种语言。有关更多信息，请参考 Dasher 网站，其中提供了完整文档、演示及培训文本。请参见 <http://www.inference.phy.cam.ac.uk/dasher/>。

17.7 查错

虚拟键盘不显示在登录屏幕上

有时，虚拟键盘不会显示在登录屏幕上。要解决此问题，请按 **Ctrl + Alt + <—** 或按 Tablet PC 上的相应键（如果使用无集成键盘的 slate 模型）重新启动 X 服务器。如果虚拟键盘仍然未显示，请将外部键盘连接到 slate 模型并使用硬件键盘登录。

Wacom 图形手写板的方向未更改

使用 `xrandr` 命令，可以在外壳中更改显示器方向。输入 `xrandr --help` 可查看可用选项。要同时更改图形手写板方向，需要按以下描述修改命令：

- 对于常规方向（0°方向）：

```
xrandr --output LVDS ---rotate normal && xsetwacom set "Mouse[7]" Rotate NONE
```

- 对于 90°旋转（顺时针，纵向）：

```
xrandr --output LVDS ---rotate right && xsetwacom set "Mouse[7]" Rotate  
CW
```

- 对于 180° 旋转（横向）：

```
xrandr --output LVDS --rotate inverted && xsetwacom set "Mouse[7]"  
Rotate HALF
```

- 对于 270° 旋转（逆时针，纵向）：

```
xrandr --output LVDS --rotate left && xsetwacom set "Mouse[7]" Rotate  
CCW
```

请注意以上命令是根据 `/etc/X11/xorg.conf` 配置文件内容而定的。如果已经使用 SaX2 按照第 17.2 节“配置手写板设备”[175]中所述配置了设备，命令将正常运行。如果手动更改了 `xorg.conf` 中手写板手写输入笔输入设备的 Identifier，请将 `"Mouse[7]"` 替换为新的 Identifier。如果您有一个具有触板支持的 Wacom 设备（您可以在手写板上使用手指移动光标），您还需要旋转触摸设备。

17.8 更多信息

这里提到的一些应用程序不提供集成联机帮助，但是您可以在 `/usr/share/doc/package/packageName` 或以下站点找到关于您所安装系统的用法和配置的实用信息：

- 有关 Xournal 手册，请参见 <http://xournal.sourceforge.net/manual.html>
- Jarnal 文档位于 <http://www.dklevine.com/general/software/tc1000/jarnal.htm#documentation>
- 有关 xstroke 手册页，请参见 <http://davesource.com/Projects/xstroke/xstroke.txt>
- 有关配置 X 的 HOWTO，请参见 Linux Wacom 网站：<http://linuxwacom.sourceforge.net/index.php/howto/x11>
- 有关 Dasher 项目详细信息的网站，请参见 <http://www.inference.phy.cam.ac.uk/dasher/>

- 有关 CellWriter 的更多信息和文档，请参见 <http://risujin.org/cellwriter/>。
- 有关 gnome-display-properties 的信息，请参见 <http://en.opensuse.org/GNOME/Multiscreen>

部分 IV. 服务

基本联网知识

Linux 提供集成进各类网络结构中所需的联网工具和功能。TCP/IP 是 Linux 惯用的协议，具有多种服务和特殊功能，本章将对此进行介绍。使用网卡、调制解调器或其他设备的网络访问可以通过 YaST 来配置。也可以手动进行配置。在本章中，仅描述基础机制和相关网络配置文件。

Linux 和其他 Unix 操作系统均使用 TCP/IP 协议。该协议不是单个网络协议，而是提供多种服务的一系列网络协议。中所列的协议专用于在两台计算机之间通过 TCP/IP 交换数据。**表 18.1 “TCP/IP 系列协议中的若干协议”** [187] 由 TCP/IP 连接而成的网络构成了全球网络，也称作“因特网”。

RFC 代表*注释请求*。RFC 由一些文档组成，用来描述各种因特网协议和操作系统及其应用程序的实施过程。RFC 文档用来描述如何设置因特网协议。要进一步了解某个协议，请参见相应的 RFC 文档。这些文档可在<http://www.ietf.org/rfc.html> 获得。

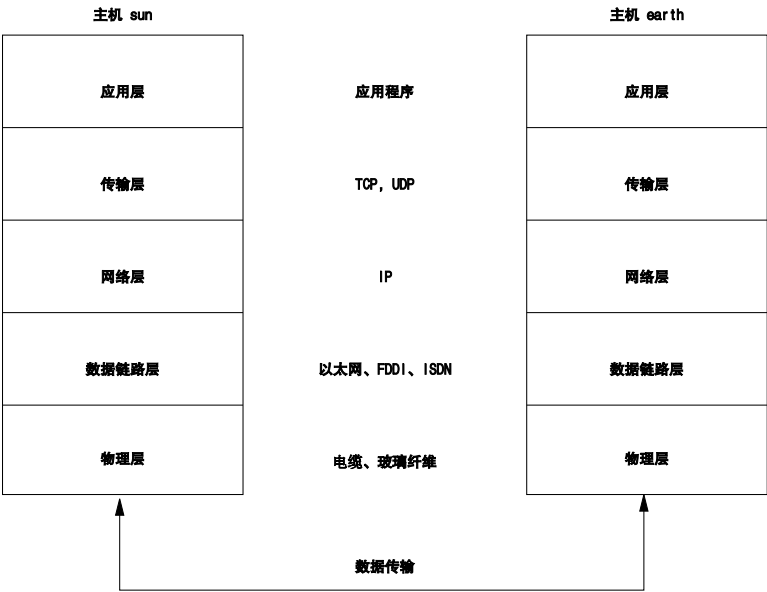
表 18.1 TCP/IP 系列协议中的若干协议

协议	描述
TCP	传送控制协议：面向连接的安全协议。要传输的数据首先由应用程序作为数据流发送，然后由操作系统转换为相应的格式。数据到达目标主机上的相应应用程序时采用最初发送时的原始数据流格式。TCP 确定传输过程中是否丢失任何数据或数据的顺序是否被打乱。只要涉及到数据序列就会实施 TCP。

协议	描述
UDP	用户数据报协议：无连接、不安全的协议。要传送的数据以应用程序生成的数据包的形式发送。不能保证数据以正确的顺序到达接收方，也可能丢失数据。 UDP 适用于面向记录的应用程序。它的等待时间比 TCP 稍短。
ICMP	因特网控制消息协议：这实际上不是一个面向最终用户的协议，而是一个特殊的控制协议，用来发出错误报告，还可以控制参与 TCP/IP 数据传送的计算机的行为。此外，它还提供一种特殊的回应方式，可以通过 ping 程序查看该方式。
IGMP	因特网组管理协议：此协议控制实施 IP 多路广播时的计算机行为。

如图 18.1 “TCP/IP 的简化层次模型”[188]中所示，数据交换在不同的层中进行。实际的网络层是通过 **IP**（因特网协议）的不安全数据传送。**IP** 的上面是 **TCP**（传送控制协议），它能够确保一定程度的数据传送安全性。**IP** 层又受底层硬件相关协议（例如以太网）的支持。

图 18.1 TCP/IP 的简化层次模型



该图为每一层都提供了一到两个示例。层次按照抽象程度排序。最底层非常接近硬件。最上层则几乎就是硬件的完全抽象化。每一层都有自己的特殊功能。每一层的特殊功能多隐含在其描述中。数据链路层和物理层表示所用的物理网络（如以太网）。

几乎所有硬件协议都在面向数据包的基础上发挥作用。因为无法一次传送所有数据，所以要将这些数据封装在包中。TCP/IP 包最大约为 64 KB。通常的包还要小得多，因为可能还要受到网络硬件的限制。以太网上的数据包最大约为 1500 字节。通过以太网发送数据时，TCP/IP 包不能超过这个限额。如果传送更多数据，操作系统需要发送更多的数据包。

为使层实现其指定功能，必须在数据包中保存与每层相关的附加信息。这些信息保存在数据包的报头中。每一层都在每个新包的开头附加一小块称为协议报头的数据。演示了一个通过以太网电缆传送的示例 TCP/IP 数据包。图 18.2 “TCP/IP 以太网包”[189] 校验和位于包的末尾而不是开头，这样更便于网络硬件处理。

图 18.2 TCP/IP 以太网包



当应用程序通过网络发送数据时，数据会穿越每个层次，所有传递都在 Linux 内核中实施（只有物理层除外）。每一层都负责准备好数据，以便传递到下一层。最底层最后负责发送数据。接收数据时则逆向执行整个过程。正像剥洋葱皮那样，在每一层中都要从传输数据中删除协议报头。最后，传输层负责使数据可供目标上的应用程序使用。通过这种方式，每一层只与其上一层或下一层通讯。对于应用程序，无论数据是通过 100 Mbit/s（兆位/秒）的 FDDI 网络传送还是通过 56 Kbit/s（千位/秒）的调制解调器线路传送，都与此无关。同样，只要数据包的格式正确，传送哪种数据对数据线也无关紧要。

18.1 IP 地址和路由

各节的论述仅限于 IPv4 网络。有关 IPv6 协议（IPv4 的后续协议）的信息，请参见 [第 18.2 节“IPv6 — 下一代的因特网”](#) [192]。

18.1.1 IP 地址

因特网上的每台计算机都有一个唯一的 32 位地址。这些 32 位（或 4 字节）地址通常按 [例 18.1“编写 IP 地址”](#) [190] 的第二行所示的格式书写。

例 18.1 编写 IP 地址

```
IP Address (binary): 11000000 10101000 00000000 00010100
IP Address (decimal): 192. 168. 0. 20
```

在十进制格式中，四字节以十进制数书写，其间以句点分隔。IP 地址被指派给主机或网络接口。除此之外不能用在其他任何地方。这条规则也有例外，但这些例外与以下消息无关。

IP 地址中的点表示分级系统。直到 20 世纪 90 年代，IP 地址仍然有严格的分类。不过，这个系统经证实太过死板，已经废止。现已改为使用无类别路由（CIDR，无类别域间路由）。

18.1.2 网络掩码和路由

网络掩码用于定义子网的地址范围。如果两台主机在同一个子网中，则它们可直接相互访问，如果不在同一个子网中，则需要网关地址，它处理子网和其他网络之间的所有流量。要检查两个 IP 地址是否位于同一个子网中，只需分别将两个地址与网络掩码进行“AND”操作。“”如果结果相同，则两个 IP 地址在同一个本地网络中。如果结果不同，则仅能通过网关连接远程 IP 地址和远程接口。

要了解网络掩码如何工作，可查看 [例 18.2“将 IP 地址链接到网络掩码”](#) [191]。网络掩码有 32 位，它确定属于网络的 IP 地址是多少。对于所有为 1 的位，将它们在 IP 地址中的相应位标记为属于网络。对于所有为 0 的位，标记为属于子网。这意味着为 1 的位越多，子网就越小。因为网络掩码总是由多个连续的 1 位组成，所以也可仅计算网络掩码中的位数。在 [例 18.2“将 IP 地址链接到网络掩码”](#) [191] 中，第一个 24 位也可写作 192.168.0.0/24。

例 18.2 将 IP 地址链接到网络掩码

```
IP address (192.168.0.20):  11000000 10101000 00000000 00010100
Netmask   (255.255.255.0):  11111111 11111111 11111111 00000000
-----
Result of the link:         11000000 10101000 00000000 00000000
In the decimal system:      192.      168.        0.        0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask   (255.255.255.0):  11111111 11111111 11111111 00000000
-----
Result of the link:         11010101 10111111 00001111 00000000
In the decimal system:      213.      95.       15.        0
```

再举个例子：通过同一以太网电缆相连的所有计算机通常都位于同一子网中，可直接访问。即使用交换机或网桥物理分隔该子网，这些主机仍然可以直接访问。

仅在网关是为目标网络配的时，才能访问本地子网外部的 IP 地址。通常情况下，只有一个网关处理所有外部流量。然而，也可能为不同的子网配置多个网关。

如果配置了网关，所有的外部 IP 包将发送到相应的网关。此网关随后会尝试以相同的方式转发该包（从主机到主机）直到到达目标主机或超过该包的 TTL（存活时间）。

表 18.2 特定地址

地址类型	描述
基本网络地址	这是网络掩码和该网络中的任意地址，如 例 18.2 “将 IP 地址链接到网络掩码” [191] 中的 Result（结果）所示。不能将此地址指派给任何主机。
广播地址	这大体表示“访问此子网内的所有主机”。要生成此地址，需要将网络掩码反转为二进制格式，并使用逻辑 OR 链接到基本网络地址。因此，以上示例会生成 192.168.0.255。该地址无法指派给任何主机。
本地主机	地址 127.0.0.1 指派给每台主机的“回路设备”。“”可以使用此地址与您自己的计算机建立连接。

由于IP地址必须在全球范围内唯一，您不能随机选择地址。共有三个地址域可用于建立基于IP的专用网络。这些地址无法与因特网上的其他地址建立任何连接，因为它们不能通过因特网传送。这些地址域在 RFC 1597 中指定，并且列在表 18.3 “专用 IP 地址域” [192]中。

表 18.3 专用 IP 地址域

网络/网络掩码	域
10.0.0.0/255.0.0.0	10.x.x.x
172.16.0.0/255.240.0.0	172.16.x.x – 172.31.x.x
192.168.0.0/255.255.0.0	192.168.x.x

18.2 IPv6 — 下一代的因特网

由于WWW（万维网）的出现，过去十五年中越来越多的计算机开始通过TCP/IP通讯，这使因特网有了突飞猛进的发展。自从1990年在CERN (<http://public.web.cern.ch>) 任职的 Tim Berners-Lee 开创了 WWW，因特网主机的数量已从几千台猛增至上亿台。

如上所述，IPv4 地址只有 32 位。而且还有不少 IP 地址丢失 - 因网络组织结构的原因而无法使用。子网中可用的地址数量是位数的平方减 2。举例来说，某个子网可以有 2 个、6 个或 14 个可用地址。如果要将 128 台主机连接到因特网，您的子网要提供 256 个 IP 地址，其中只有 254 个可用，因为有两个 IP 地址需要供该子网本身的结构使用：广播和基础网络地址。

在当前的 IPv4 协议下，DHCP 或 NAT（网络地址转换）是用来避免出现地址短缺的典型机制。这些方法与用来分隔专用地址空间和公用地址空间的规定相结合，肯定能够缓解短缺状况；它们的问题在于不仅配置烦琐，而且也加重了维护的负担。要在 IPv4 网络内设置主机，您需要若干地址项，如主机本身的 IP 地址、子网掩码、网关地址，可能还要提供名称服务器地址。所有这些项都是必需的，而且无法从其他任何地方得到这些项。

利用 IPv6，地址的短缺和复杂的配置都将成为过去。以下各节进一步描述了 IPv6 带来的改进和优点，以及如何从旧协议过渡到新协议。

18.2.1 优点

新协议中最为重要同时也最为显著的改进在于对可用地址空间的极大扩容。IPv6 地址由 128 位值而不是传统的 32 位值组成，它提供的 IP 地址数目多达 10 的 15 次方的若干倍。

不过，IPv6 与以前的不同不仅限于长度，其内部结构也发生了变化，这种结构可以包含更多的有关系统和系统所属网络的具体信息。有关详细信息，请参见第 18.2.2 节“地址类型和结构”[194]。

以下列出了新协议的其他一些优点：

自动配置

IPv6 使网络可以支持“即插即用”，这意味着无需任何手动配置即可将新安装的系统集成到（本地）网络中。新主机可以使用其自动配置机制，依赖名为邻居发现(ND)的协议从邻近的路由器提供的信息中得到自己的地址。这种方法不要求管理员参与，并且无需维护用于分配地址的中央服务器-这是 ipv4 无法媲美的（在 ipv4 中需要使用 DHCP 服务器来自动分配地址）。

移动能力

利用 IPv6，为一个网络接口同时指派多个地址成为可能。这使得用户能方便地访问几个网络，可比作手机公司提供的国际漫游服务：您携带手机出境时，手机一旦进入相应区域就会自动登录外国服务，因此无论您在哪儿，都可以用同一号码联系您，并且可以像在家乡一样拨打电话。

安全通讯

在 IPv4 中，网络安全是一项附加功能。IPv6 则将 IPsec 作为其核心功能之一，允许系统通过安全隧道通讯，避免被因特网上的外来者窃听。

向后兼容性

实际上，要想将整个因特网一下子从 IPv4 转换为 IPv6 是不可能的。因此，这两个协议不仅要能在因特网上同时存在，还应能够同时存在于一个系统中，这一点至关重要。要实现这一点，一方面两种地址应兼容（IPv4 地址可以轻松转换为 IPv6 地址），另一方面还要使用一定数量的隧道。请参见第 18.2.3 节“IPv4 与 IPv6 并存”[197]。此外，系统可以依赖双栈 IP 技术同时支持两种协议，这意味着系统中有两种完全分开的网络堆栈，从而避免这两种版本的协议相互影响。

通过多路广播的自定义服务

在 IPv4 中，有些服务（如 SMB）需要向本地网络中的所有主机广播其数据包。IPv6 则采用一种更为精确的方式，通过多路广播支持服务器对主机寻址，即对属于一组的若干主机寻址（这不同于通过广播对所有主机寻址或通过单路广播对每台主机逐个寻址）。将哪些主机作为一组来寻址可能要取决于具体的应用程序。可使用一些预定义的组来寻址，例如对所有名称服务器寻址（所有名称服务器多路广播组），或对所有路由器寻址（所有路由器多路广播组）。

18.2.2 地址类型和结构

如上所述，目前的 IP 协议在两个重要方面有缺陷：IP 地址日益短缺，配置网络、维护路由选择表的任务变得越来越复杂和艰难。IPv6 通过将地址空间扩展到 128 位解决了第一个问题。通过引入分级地址结构，结合先进的网络地址分配技术和多宿主功能（将多个地址指派给同一个设备，从而支持对多个网络的访问），第二个问题也迎刃而解。

使用 IPv6 时，了解三种类型的地址十分有用：

单路广播

这类地址只与一个网络接口关联。采用这类地址的包只传递到一个目标。因此，使用单路广播地址可以将包传送到本地网络或因特网上的单个主机。

多路广播

这类地址与一组网络接口相关。采用这类地址的包将传递到属于该组的所有目标。多路广播地址主要供特定网络服务使用，用于以有序的方式与特定的主机组通讯。

任意广播

这类地址与一组接口相关。采用这类地址的包将根据基础路由协议的原则，传递给该组中与发送方最为接近的成员。任意广播地址便于主机在特定网络区域内找到提供特定服务的服务器。同一类型的所有服务器都具有相同的任意广播地址。在请求服务时，主机会收到路由协议决定的最接近它的服务器的回复。如果出于某种原因此服务器无法回复，协议会自动选择距离稍远一些的服务器，依此类推。

IPv6 地址分为八组，每组四位数字，代表十六位，采用十六进制表示法。各组之间用冒号 (:) 分隔。可以删除某组中的前置零字节，但不能删除组中或组末的零。另一个约定是：连续的零字节若超过四个，则可以省略为双冒号形式。

不过，每个地址只允许有一个这样的 ::。中演示了这种简写表示法，其中的三行全部表示同一地址。**例 18.3 “示例 IPv6 地址”** [195]

例 18.3 示例 IPv6 地址

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :      0 :      0 :      0 :      0 : 10 : 1000 : 1a4
fe80 :                               : 10 : 1000 : 1a4
```

IPv6 地址的每个部分都有明确的功能。前面的字节构成前缀，用于指定地址类型。中间部分是地址的网络部分，但可以不用。地址的结尾构成主机部分。在 IPv6 中，网络掩码是通过在地址末尾的斜杠后指明前缀的长度来定义的。**例 18.4 “指定前缀长度的 IPv6 地址”** [195] 中的地址包含上述信息，即：前 64 位构成地址的网络部分，后 64 位构成地址的主机部分。换言之，64 表示网络掩码由左起的 64 个 1 位值构成。正如 IPv4，要用 AND 将 IP 地址与子网值结合起来，以确定主机位于同一子网中还是其他网络中。

例 18.4 指定前缀长度的 IPv6 地址

```
fe80::10:1000:1a4/64
```

IPv6 可以识别几种预定义的前缀类型。其中有些列在 **表 18.4 “各种 IPv6 前缀”** [195] 中。

表 18.4 各种 IPv6 前缀

前缀（十六进制）	定义
00	IPv4 地址和 IPv6 上的 IPv4 兼容地址。这些用于与 IPv4 保持兼容。要使用这些地址，仍然需要依赖路由器将 IPv6 包转换为 IPv4 包。有若干特殊地址（如用于回路设备的地址）也采用此前缀。
2 或 3 作为第一个数字	可聚合全局单路广播地址。类似 IPv4 的情况，可以指定某个接口作为特定子网的一部分。目前，有以下地址空间：2001::/16（生产质量地址空间）和 2002::/16（6to4 地址空间）。
fe80::/10	链路本地地址。不应路由带有这种前缀的地址，而只能从同一子网中访问。

前缀（十六进制） 定义	
fec0::/10	站点本地地址。可以路由这种地址，但只局限在它们所属的组织网络之内。实际上，这些是相当于当前的专用网络地址空间（如 10.x.x.x）的 IPv6 地址。
ff	这些是多路广播地址。

单路广播地址由三个基本部分组成：

公共拓扑结构

第一部分（也包含上述前缀之一）用于通过公共因特网路由数据包。其中包含提供因特网访问的公司或机构的相关信息。

站点拓扑结构

第二部分包含要将包传递到的子网的路由信息。

接口 ID

第三部分标识要将包传递到的接口。其中允许使用 MAC。由于 MAC 是硬件厂商编程到设备中的全球唯一的固定标识符，配置过程得到了极大简化。事实上，前 64 个地址位共同构成 EUI-64 令牌，后 48 位从 MAC 中提取，其余的 24 位包含有关令牌类型的特殊信息。这样还可以将 EUI-64 令牌指派给没有 MAC 的接口，如基于 PPP 或 ISDN 的接口。

在这个基础结构之上，IPv6 还区分五种不同的单路广播地址：

- :: (未指定)
在首次初始化接口时，即无法通过其他方法确定地址时，这类地址可用作主机的源地址。
- :::1 (回路)
回路设备的地址。

IPv4 兼容地址

IPv6 地址由 IPv4 地址和 96 个零位组成的前缀构成。这类兼容地址用于隧道通讯进程（请参见第 18.2.3 节“IPv4 与 IPv6 并存”[197]），以便 IPv4 和 IPv6 主机与在纯 IPv4 环境中操作的其他主机通讯。

映射到 IPv6 的 IPv4 地址

这类地址以 IPv6 表示法指定纯 IPv4 地址。

本地地址

有两类地址可供本地使用：

链路本地

这类地址只能在本地子网中使用。不能将源地址或目标地址采用此类地址的包路由到因特网或其他子网。这些地址包含特殊的前缀 (fe80::/10) 和网卡的接口 ID，中间部分为零字节。这类地址在自动配置过程中使用，用于与同一子网中的其他主机通讯。

站点本地

可以将采用这类地址的包路由到其他子网，但不能路由到更广阔的因特网 - 不能跨越组织自身的网络。这类地址用于内部网，相当于 IPv4 定义的专用地址空间。其中包含特殊的前缀 (fec0::/10)、接口 ID，及指定子网 ID 的 16 位域。其余部分也是零字节。

作为 IPv6 引进的全新功能，每个网络接口通常可以获得多个 IP 地址，这个功能的优点即在于：可以通过同一接口访问多个网络。其中一个网络可以使用 MAC 和已知前缀进行完全的自动配置，这样一启用 IPv6（使用链路本地地址），即可访问本地网络中的所有主机。由于其中使用了 MAC，所用的任何 IP 地址都是全球唯一的。地址中只有指定站点拓扑结构和公共拓扑结构的部分才是可变部分，这取决于主机当前运行所在的实际网络。

要使主机在不同网络间切换，主机至少需要两个地址。其中之一 - 本地地址，不仅包含接口 ID 而且包含该主机通常所属的本地网络的标识符（以及相应的前缀）。本地地址是静态地址，因此一般不变。所有要发送到移动主机的包仍可传递到该主机，不管它是在本地网络还是其他任何网络中操作。这一点得益于 IPv6 引进的全新功能，如无状态自动配置和邻居发现。除本地地址之外，移动主机还获得一个或多个额外的地址，这些地址属于该主机漫游到的外地网络。这些地址称为转交地址。本地网络有一种功能，可以在主机漫游到外地时转发要发送给该主机的所有包。在 IPv6 环境中，这项任务由本地代理来完成，该代理可以接收要发送到本地地址的所有包，并通过隧道进行转发。另一方面，发送到转交地址的那些包可直接转发到移动主机，而不必进行任何特殊的迂回处理。

18.2.3 IPv4 与 IPv6 并存

将与因特网相连的所有主机从 IPv4 迁移到 IPv6 是一个逐步的过程。这两种协议将在未来一定时间内并存。通过双栈技术来实施这两种协议，可以在同一系统上同时支持这两种协议。但这仍然没有解决支持 IPv6 的主机如何与 IPv4 主

机通讯，以及应如何通过当前网络（主要基于 IPv4）传输 IPv6 包的问题。最好的解决方案就是提供隧道处理功能和兼容地址（请参见 第 18.2.2 节“地址类型和结构”[194]）。

IPv6 主机多少孤立于（全球）IPv4 网络，它可通过隧道通讯：IPv6 包封装为 IPv4 包，以便在 IPv4 网络中移动。这种在两个 IPv4 主机间的连接被称为隧道。要实现这种功能，包必须包含 IPv6 目标地址（或相应的前缀），以及隧道接收端的远程主机的 IPv4 地址。根据主机管理员间的协议，可以手动配置基本的隧道。这也称作静态隧道。

但是，静态隧道的配置和维护往往过于烦琐，不能适应日常通讯需要。因此，IPv6 提供了三种不同的动态隧道方法：

6over4

IPv6 包被自动封装为 IPv4 包，并通过支持多路广播的 IPv4 网络发送。这种方法诱导 IPv6 将整个网络（因特网）视为一个巨大的局域网 (LAN)。这样即可自动确定 IPv4 隧道的接收端。不过，这种方法不够灵活，并且还因为 IP 多路广播在因特网上尚未普及而不易推行。因此，它提供的解决方案仅适用于支持多路广播的小型公司网络或机构网络。RFC 2529 中对这种方法作出了规定。

6to4

利用这种方法，可以从 IPv6 地址自动生成 IPv4 地址，从而支持孤立的 IPv6 主机通过 IPv4 网络进行通讯。不过，用这种方法在孤立的 IPv6 主机和因特网之间通讯时存在一些问题。RFC 3056 中对这种方法进行了描述。

IPv6 隧道中介程序

这种方法依赖特殊的服务器为 IPv6 主机提供专用隧道。RFC 3053 中对此进行了描述。

18.2.4 配置 IPv6

要配置 IPv6，通常无需在各个工作stations上执行任何更改。默认情况下启用 IPv6。安装期间，您可以在“Network Configuration”一节（第 4 章 *Installation with YaST*, ↑部署指南）中所述的网络配置步骤中禁用它。要在已安装系统上禁用或启用 IPv6，请使用 YaST 网络设置模块。在全局设置选项卡上，根据需要选中或取消选中启用 IPv6 选项。要手动启用 IPv6，请以 root 身份输入 `modprobe ipv6`。

由于 IPv6 使用自动配置，将给网卡指派链路-本地网络中的地址。一般不在工作站上管理路由选择表。工作站可以使用路由器广告协议查询网络路由器，了解应实施的前缀和网关。使用 radvd 程序可以设置 IPv6 路由器。此程序会通知工作站对 IPv6 地址使用哪个前缀和哪个路由器。或者，可以使用 zebra/quagga 自动配置两个地址和路由选择。

有关如何使用 `/etc/sysconfig/network` 文件设置各种隧道的信息，请参见 `ifcfg-tunnel (5)` 手册页。

18.2.5 有关详细信息

上文的概述中并未全面论述 IPv6 这一主题。有关这种新协议的深入讨论，请参见以下联机文档和书目：

<http://www.ipv6.org/>

学习 IPv6 知识的起点。

<http://www.ipv6day.org>

启动您自己的 IPv6 网络所需的所有信息。

<http://www.ipv6-to-standard.org/>

已启用 IPv6 的产品列表。

<http://www.bieringer.de/linux/IPv6/>

在此可找到 Linux IPv6-HOWTO 以及许多与该主题有关的链接。

RFC 2640

有关 IPv6 的基础 RFC。

IPv6 Essentials

Silvia Hagen 所著的 *IPv6 Essentials*(ISBN 0-596-00125-8) 中描述了该主题的所有重要方面。

18.3 名称解析

DNS 有助于将 IP 地址指派给一个或多个名称，并将名称指派给 IP 地址。在 Linux 中，这种转换通常由一种特殊的称为 bind 的软件来完成。负责这种转换

的计算机称为名称服务器。这些名称构成了分级系统，各个名称组成部分之间用圆点分隔。不过，这个名称层次与上述 IP 地址层次无关。

考虑以 `hostname.domain` 格式书写的完整名称，如 `jupiter.example.com`。完整名称，即完全限定的域名 (fqdn)，由主机名和域名 (`example.com`) 组成。后者还包含顶级域或 TLD (`com`)。

TLD 的指派由于历史原因已经变得十分混乱。传统的指派方法是美国所用的三字母域名，而世界其他地方采用的标准是双字母 ISO 国家/地区代码。此外，2000 年还引进了较长的 TLD，表示特定的活动领域（例如 `.info`、`.name` 和 `.museum`）。

在因特网发展的早期阶段（1990 年之前），文件 `/etc/hosts` 被用来储存因特网上表示的所有计算机的名称。后来事实证明随着接入因特网的计算机与日俱增，这种方法很快就行不通了。为此人们开发了一个分散式数据库，以十分分散的方式储存主机名。这个数据库类似名称服务器，它并不储存与因特网上的所有主机相关的数据，但可以向其他名称服务器发送请求。

位于层次顶级的是根名称服务器。这些根名称服务器管理顶级域，并由网络信息中心 (NIC) 运行。每个根名称服务器都了解负责特定顶级域的名称服务器。有关顶级域 NIC 的信息，请参见 <http://www.internic.net>。

DNS 不仅可以解析主机名，还能够为整个域识别出负责接收整个域的电子邮件的主机 - 邮件交换器 (MX)。

为解析 IP 地址，您的计算机必须了解至少一个名称服务器及其 IP 地址。借助 YaST 可以轻松指定这样的名称服务器。如果建立的是调制解调器拨号连接，则根本无需手动配置名称服务器。拨号协议可以在建立连接后提供名称服务器的地址。

whois 协议与 DNS 密切相关。使用此程序可以快速找出负责特定域的服务器。

注意: MDNS 和 `.local` 域名

`.local` 顶级域由解析程序视为 `link-local` 域。DNS 请求作为多路广播 DNS 请求（而不是常规 DNS 请求）发送。如果已在名称服务器配置中使用 `.local` 域，必须在 `/etc/host.conf` 中关闭此选项。另请阅读 `host.conf` 手册页。

如果要在安装期间关闭 MDNS，请使用 `nomdns=1` 作为引导参数。

有关多路广播 DNS 的详细信息，请参见 <http://www.multicastdns.org>。

18.4 使用 YaST 配置网络连接

Linux 上有多个支持的联网类型。其中多数使用不同的设备名，配置文件分布在文件系统上的多个位置。关于手动网络配置方面的详细概述，请参见第 18.6 节“[手动配置网络连接](#)”[219]。

在 SUSE Linux Enterprise Desktop 上（NetworkManager 默认已激活），所有网卡均已配置。如果 NetworkManager 未激活，仅具有链路的第一个接口（已连接网络电缆）会自动配置。可随时在已安装系统中配置额外的硬件。以下各节将介绍 SUSE Linux Enterprise Desktop 支持的所有网络连接类型的网络配置。

18.4.1 使用 YaST 配置网卡

要在 YaST 中配置有线或无线网卡，请选择 **网络设备 > 网络设置**。启动模块后，YaST 显示 **网络设置** 对话框，带有四个选项卡：**全局选项**、**概述**、**主机名/DNS** 和 **路由选择**。

通过 **全局选项** 选项卡可设置常规联网选项，如 NetworkManager 的使用、IPv6 和 DHCP 常规选项。有关详细信息，请参见“[配置全局联网选项](#)”一节 [202]。

概述 选项卡包含关于已安装网络接口和配置的信息。会列出已正确检测到的所有网卡及其名称。您可在此对话框中手动配置新卡、删除或更改其配置。如果要手动配置未自动检测到的卡，请参见“[配置未检测到的网卡](#)”一节 [208]。如果要更改已配置卡的配置，请参见“[更改网卡的配置](#)”一节 [203]。

通过 **主机名/DNS** 选项卡可设置计算机的主机名和要使用的服务器名称。有关详细信息，请参见“[配置主机名和 DNS](#)”一节 [208]。

路由选择 选项卡用于配置路由选择。有关更多信息，请参见“[配置路由选择](#)”一节 [210]。

图 18.3 配置网络设置



配置全局联网选项

通过 YaST 网络设置模块的全局选项选项卡可设置重要的全局联网选项，如 NetworkManager 的使用、IPv6 和 DHCP 客户端选项。这些设置适用于所有网络接口。

在网络设置方法中，选择管理网络连接的方法。如果希望 NetworkManager 桌面小程序管理所有接口的连接，则选择通过 *NetworkManager* 的用户控制方法。此选项非常适用于在多个有线和无线网络之间切换。如果您没有运行桌面环境（GNOME 或 KDE）或者您的计算机是 Xen 服务器、虚拟系统，又或在网络中提供网络服务（如 DHCP 或 DNS），请使用通过 *ifup* 的传统方法。如果使用 NetworkManager，则应使用 nm-applet 配置网络选项，并且网络设置模块的概述、主机名/DNS 和路由选择选项卡会被禁用。有关 NetworkManager 的更多信息，请参见第 22 章 使用 *NetworkManager* [255]。

在 IPv6 协议设置中，选择是否希望使用 IPv6 协议。可将 IPv6 与 IPv4 一起使用。默认情况下激活 IPv6。但是在不使用 IPv6 协议的网络中，如果禁用 IPv6 协议，响应时间会更快。如果要禁用 IPv6，请取消选中启用 IPv6 选项。这样将为 IPv6 禁用内核模块自动加载。这将在重引导后应用。

在 *DHCP* 客户端选项中，配置 *DHCP* 客户端的选项。如果希望 *DHCP* 客户端要求服务器始终广播其响应，则选中 *请求广播响应*。如果您的计算机在不同网络之间移动，则可能需要选中此选项。在单个网络上，每个 *DHCP* 客户端的 *DHCP* 客户端标识符必须不同。如果保留为空，会默认为网络接口的硬件地址。但是，如果正在运行若干使用相同网络接口（即相同硬件地址）的虚拟机，则在此处指定唯一的自由格式标识符。

要发送的主机名指定 *dhcpcd* 向 *DHCP* 服务器发送消息时用于主机名选项的字符串。某些 *DHCP* 服务器会根据此主机名（动态 DNS）来更新名称服务器区域（转发和反转记录）。此外，有些 *DHCP* 服务器要求要发送的主机名选项字段包含来自客户端的 *DHCP* 消息中的特定字符串。选择 *AUTO* 可发送当前主机名（即 */etc/HOSTNAME* 中定义的主机名）。将选项字段保留为空即可不发送任何主机名。如果您不希望根据 *DHCP* 中的信息更改默认路由，则取消选中 *通过 DHCP 更改默认路由*。

更改网卡的配置

要更改网卡的配置，请在 *YaST 网络设置 > 概述* 中已检测到的网卡列表中选择一张网卡，然后单击 *编辑*。将显示 *网卡设置* 对话框，可在其中使用 *常规*、*地址* 和 *硬件* 选项卡调整网卡配置。有关无线网卡配置的信息，请参见第 19.1.2 节“*用 YaST 配置*” [238]。

配置 IP 地址

您可在 *网卡设置* 对话框的 *地址* 选项卡中设置网卡的 IP 地址或 IP 地址的确定方法。同时支持 IPv4 和 IPv6 地址。网卡可设置为 *无 IP 地址*（对于绑定设备很有用）、*静态指派的 IP 地址*（IPv4 或 IPv6）或通过 *DHCP* 和/或 *Zeroconf* 指派的 *动态地址*。

如果使用 *动态地址*，则选择是使用仅 *DHCP V4*（用于 DHCPv4）、*DHCP V6*（用于 DHCPv6）还是 *DHCP V4 和 V6*。

如果可能，安装期间的首个带链接的可用网卡将会通过 *DHCP* 自动配置为使用自动 IP 地址。在 SUSE Linux Enterprise Desktop 上（*NetworkManager* 默认已激活），所有网卡均已配置。

如果使用的是 DSL 线路，但 ISP（因特网服务提供商）没有指派静态 IP，此时还应使用 *DHCP*。如果决定使用 *DHCP*，则在 *YaST 网卡配置模块的网络设置* 对话框 *全局选项* 选项卡的 *DHCP 客户端选项* 中配置细节。在 *请求广播响应* 中指定 *DHCP* 客户端是否应要求服务器始终广播其响应。如果您的计算机是在不同网

络间移动的移动客户端，则可能需要使用此选项。如果您使用虚拟主机设置，其中不同的主机都通过同一接口通信，则需要用 *DHCP 客户端标识符* 来区分。

DHCP 比较适合客户机配置，但不太适合服务器配置。要设置静态 IP 地址，请如下继续操作：

- 1 在 YaST 网卡配置模块的概述选项卡的已检测到网卡列表中选择一张网卡，然后单击 *编辑*。
- 2 在地址选项卡中，选择静态指派的 IP 地址。
- 3 输入 IP 地址。IPv4 和 IPv6 地址均可使用。在子网掩码中输入子网掩码。如果使用 IPv6 地址，则对于前缀长度使用 /64 格式的子网掩码。

或者，您可以为此地址输入一个完全限定的主机名，该主机名将写入到 `/etc/hosts` 配置文件。
- 4 单击 *下一步*。
- 5 要激活配置，请单击 *确定*。

如果使用静态地址，则不会自动配置名称服务器和默认网关。要配置名称服务器，请按照“配置主机名和 DNS”一节 [208] 中所述进行。要配置网关，请按照“配置路由选择”一节 [210] 中所述进行。

配置别名

一个网络设备可以有多个 IP 地址，称为别名。要为网络设置别名，请如下继续操作：

- 1 在 YaST 网卡配置模块的概述选项卡的已检测到网卡列表中选择一张网卡，然后单击 *编辑*。
- 2 在地址 > 附加地址选项卡中，单击 *添加*。
- 3 输入别名、IP 地址和网络掩码。不要在别名中包含接口名称。
- 4 单击 *确定*。
- 5 单击 *下一步*。

- 6 要激活配置，请单击**确定**。

更改设备名称和 Udev 规则

可更改网卡在使用时的设备名称。还可确定 udev 是通过网卡的硬件 (MAC) 地址还是通过总线 ID 来标识网卡。在大型服务器中，更偏向于使用后者，以便减少网卡的热交换。要使用 YaST 来设置这些选项，请如下继续操作：

- 1 在 YaST 网络设置模块的概述选项卡的已检测到网卡列表中选择一个网卡，然后单击**编辑**。
- 2 转到**硬件**选项卡。当前设备名称显示在 *Udev 规则* 中。单击**更改**。
- 3 选择 udev 应通过网卡的 *MAC 地址* 还是 *总线 ID* 来识别网卡。网卡的当前 MAC 地址和总线 ID 显示在对话框中。
- 4 要更改设备名称，请选中**更改设备名称**选项并编辑名称。
- 5 单击**确定**然后单击**下一步**。
- 6 要激活配置，请单击**确定**。

更改网卡内核驱动程序

对于某些网卡，可能会提供某些内核驱动程序。如果网卡已配置，YaST 允许从可用的合适驱动程序列表中选择一个要使用的内核驱动程序。还可为内核驱动程序指定选项。要使用 YaST 来设置这些选项，请如下继续操作：

- 1 在 YaST 网络设置模块的概述选项卡的已检测到网卡列表中选择一个网卡，然后单击**编辑**。
- 2 转到**硬件**选项卡。
- 3 在**模块名称**中选择要使用的内核驱动程序。在**选项**中为所选驱动程序输入所有选项，形式为 *option=value* 。如果使用多个选项，应用空格分隔这些选项。
- 4 单击**确定**然后单击**下一步**。
- 5 要激活配置，请单击**确定**。

激活网络设备

如果您使用通过 `ifup` 的传统方法，则可以配置为在引导期间、连接电缆后或检测到网卡后启动设备，也可以配置为手动或从不启动设备。要更改设备启动，请如下继续操作：

- 1 在 YaST 中，在 *网络设备 > 网络设置* 中的已检测网卡列表中选择一张网卡，然后单击 *编辑*。
- 2 在 *常规选项卡* 中，从 *设备激活* 选择所希望的项。

选择在 *引导时* 可在系统引导时启动设备。使用在 *电缆连接时* 将对任何现有物理连接监视接口。使用在 *热插拔时* 将在接口可用时立即设置。这与在 *引导时* 选项很相似，唯一区别是如果引导时不存在接口，不会发生错误。选择 *手动* 可通过 `ifup` 或 `KInternet` 手动控制接口。选择 *从不* 将不启动设备。通过 *NFS 根* 与在 *引导时* 相似，但使用 `rcnetwork stop` 命令时接口不会关闭。如果您使用 `nfs` 或 `iscsi root` 文件系统，则选择此选项。

- 3 单击 *下一步*。
- 4 要激活配置，请单击 *确定*。

通常，仅系统管理员可以激活或停用网络接口。如果希望任何用户均能通过 `KInternet` 激活此接口，请选择 *通过 Kinternet 为非 root 用户启用设备控制*。

设置最大传输单位大小

您可为接口设置最大传输单位 (MTU)。MTU 是指允许的最大包大小（以字节为单位）。更高的 MTU 可带来更高的带宽效率。但是较大的包有时可能会堵塞较慢的接口，从而增加后续包的延迟。

- 1 在 YaST 中，在 *网络设备 > 网络设置* 中的已检测网卡列表中选择一张网卡，然后单击 *编辑*。
- 2 在 *常规选项卡* 中，从 *设置 MTU* 列表中选择所需项。
- 3 单击 *下一步*。
- 4 要激活配置，请单击 *确定*。

配置防火墙

无须输入详细的防火墙设置（如第 9.4.1 节“Configuring the Firewall with YaST”（第 9 章 *Masquerading and Firewalls*, ↑安全指南）中所述），您就能在设备设置过程中确定设备的基本防火墙设置。按如下所示继续：

- 1 打开 YaST 网络设备 > 网络设置模块。在概述选项卡中，从已检测到的网卡列表中选择一个网卡，然后单击编辑。
- 2 进入网络设置对话框的常规选项卡。
- 3 确定应指派接口的防火墙区域。下列选项可用：

防火墙已禁用

此选项只有在禁用防火墙和防火墙未在运行时才可用。仅当计算机属于受外置防火墙保护的大型网络时才使用此选项。

自动指派区域

此选项只有在启用防火墙后才可用。防火墙正在运行且接口自动指派给防火墙区域。包含关键字 any 的区域或外部区域将用于此类接口。

内部区域（未保护）

防火墙正在运行，但不会强制执行任何规则来保护此接口。仅当计算机属于受外置防火墙保护的大型网络时才使用此选项。当计算机具有多个网络接口时，此选项还可用于连接到内部网络的接口。

隔离区域

隔离区域是位于内部网络和（恶意）因特网之前的附加防线。可从内部网络和因特网访问指派到此区域的主机，但指派到此区域的主机无法访问内部网络。

外部区域

防火墙在此接口上运行并且全面保护其不受其他（假设为恶意）网络流量影响。这是默认选项。

- 4 单击下一步。
- 5 单击确定即可激活配置。

配置未检测到的网卡

可能未正确检测到您的网卡。在此情况下，已检测到网卡列表中不会包含此网卡。如果确定系统包含网卡的驱动程序，则可以手动对其进行配置。还可以配置特殊网络设备类型，例如网桥、绑定、TUN 或 TAP。要配置未检测到的网卡或特殊设备，请如下操作：

- 1 在 YaST 中的 *网络设备 > 网络设置 > 概述* 对话框中，单击 **添加**。
- 2 在 *硬件* 对话框中，从可用选项中设置接口的 **设备类型** 和 **配置名称**。如果网卡为 PCMCIA 或 USB 设备，则激活相应的复选框，并选择下一步退出此对话框。或者，如果需要，您可定义要用于网卡的 **内核模块名称** 及其选项。
- 3 单击 **下一步**。
- 4 在 *常规*、*地址* 和 *硬件* 选项卡中，配置所有所需的选项，如接口的 IP 地址、设备激活或防火墙区域。有关配置选项的更多信息，请参见“**更改网卡的配置**”一节 [203]。
- 5 如果选择 **无线** 作为接口的设备类型，则在下一个对话框中配置无线连接。有关无线设备配置的详细信息可在 **第 19.1 节 “无线 LAN”** [235] 中获得。
- 6 单击 **下一步**。
- 7 要激活新网络配置，请单击 **确定**。

配置主机名和 DNS

如果安装期间没有更改网络配置并且有线网卡已经可用，则已为计算机自动生成主机名并且已激活 DHCP。这同样适用于主机连接到网络环境所需的名称服务信息。如果网络地址设置使用了 DHCP，则会向域名服务器列表自动填充相应数据。如果希望使用静态设置，则手动设置这些值。

要更改计算机名称并调整名称服务器搜索列表，则如下继续操作：

- 1 转到 YaST 中的 *网络设备* 模块中的 *网络设置 > 主机名/DNS* 选项卡。
- 2 输入 **主机名**，如果需要，也输入 **域名**。如果此计算机是邮件服务器，则该域特别重要。请注意，主机名是全局性的，应用于所有网络接口。

如果使用 DHCP 获取 IP 地址，则计算机的主机名将由 DHCP 自动设置。如果您连接到不同网络，您可能希望禁用此行为，因为它们可能指派不同主机名，而在运行时更改主机名可能会导致图形桌面混乱。要禁用使用 DHCP 获取 IP 地址，请取消选中通过 *DHCP* 更改主机名。

如果要使用 DHCP 获取 IP 地址，您的主机名将会默认写入到 `/etc/hosts` 并且可以作为 `127.0.0.2` IP 地址解析。要禁用它，请取消选中将主机名写到 `/etc/hosts`，但是请注意，您的主机名在没有活动网络的情况下将不可解析。

- 3 在修改 DNS 配置中，请选择修改 DNS 配置（名称服务器、搜索列表以及 `/etc/resolv.conf` 文件的内容）的方式。

如果选择了使用默认策略选项，则配置由 `netconfig` 脚本处理，该脚本合并了静态定义的数据（通过 YaST 或在配置文件中）与动态获取的数据（来自 DHCP 客户端或 NetworkManager）。在大多数情况下，此默认策略就足够了。

如果选择了仅手动选项，则不允许 `netconfig` 修改 `/etc/resolv.conf` 文件。但是，此文件可手动编辑。

如果已选择自定义策略选项，则应指定用于定义合并策略的自定义策略规则字符串。该字符串包含了接口名称的逗号分隔列表，可考虑作为设置的有效源。除了完整接口名称，还允许使用可匹配多个接口的基本通配符。例如，`eth* ppp?` 首先以所有 `eth` 为目标，然后是 `ppp0` 到 `ppp9` 的所有接口。有两个特殊策略值表示如何应用 `/etc/sysconfig/network/config` 文件中定义的静态设置：

STATIC

静态设置必须与动态设置合并到一起。

STATIC_FALLBACK

仅当动态配置不可用时，才使用静态设置。

有关更多信息，请参见 `man 8 netconfig`。

- 4 输入名称服务器并填写域搜索列表。名称服务器必须由 IP 地址指定（如 `192.168.1.116`），而非由主机名指定。域搜索选项卡中指定的名称是用于解析主机名（无指定域）的域名。如果使用多个域搜索，则使用逗号或空格分隔域。

- 5 要激活配置，请单击**确定**。

配置路由选择

要使计算机能够与其他计算机和其他网络进行通信，必须提供路由选择信息以使网络流量使用正确的路径。如果使用 **DHCP**，则将自动提供此信息。如果使用静态设置，则必须手动添加此数据。

- 1 在 YaST 中，转到**网络设置 > 路由选择**。
- 2 输入默认网关的 IP 地址。默认网关与每个可能的目标匹配，但是如果存在任何与所需地址匹配的其他项，则使用该项代替默认路由。
- 3 可在**路由选择表**中输入多个项。输入**目标网络 IP 地址**、**网关 IP 地址**和**网络掩码**。选择将流量路由到定义的网络要经过的设备（减号代表任何设备）。要忽略所有这些值，请使用减号-。要在表中输入默认网关，请在**目标**字段中使用默认。

注意

如果使用更多的默认路由，则可以指定用于确定具有更高优先级的路由的度量选项。要指定度量选项，则在**选项**中输入 `-metricnumber`。默认情况下使用具有最高度量的路由。如果网络设备已断开连接，则删除其路由并使用下一个路由。但是，当前内核在静态路由选择中不使用度量，仅路由选择守护程序（如 `multipathd`）才在静态路由中使用度量。

- 4 如果系统是路由器，则在**网络设置**中启用 **IP 转发**选项。
- 5 要激活配置，请单击**确定**。

18.4.2 调制解调器

在 YaST 控制中心中，可以在**网络设备 > 调制解调器**下访问调制解调器配置。如果调制解调器已自动检测到，请转到**调制解调器设备**选项卡，并通过单击**添加**打开用于手动配置的对话框。在**调制解调器设备**下输入调制解调器连接到的接口。

提示: CDMA 和 GPRS 调制解调器

与配置普通调制解调器一样，用 YaST 调制解调器模块配置支持的 CDMA 和 GPRS 调制解调器。

图 18.4 调制解调器配置



调制解调器参数

输入所有的调制解调器配置值。调制解调器设备指定与调制解调器连接的端口。 ttyS0、ttyS1... [更多](#)

调制解调器设备 (U):

/dev/modem

拨号前缀 (如果需要) (O):

拨号方式

特殊设置

☒ 音频拨号 (D)

☒ 扬声器打开 (S)

☐ 脉冲拨号 (P)

☒ 检测拨号音 (E)

细节 (D)

 帮助

 取消 (C)

 后退 (B)

 下一步 (N)

如果位于专用交换分机 (PBX) 之后，则可能需要输入拨号前缀。该前缀通常是零。请参考随 PBX 附带的描述了解相关信息。同时还要选择使用音频拨号还是脉冲拨号、是否打开扬声器，以及调制解调器是否应在检测到拨号音之前一直等待。如果调制解调器连接到交换机，则不应启用最后一个选项。

在细节之下，设置波特率和调制解调器的初始化字符串。只有在调制解调器不是自动检测到的或者需要特殊设置才能传送数据时，才应更改以上设置。这种情况主要发生在 ISDN 终端适配器上。单击确定可退出此对话框。要将调制解调器的控制权委托给没有根权限的普通用户，请激活通过 Kinternet 为非 root 用户启用设备控制。这样，不具备管理员权限的用户即可激活或取消激活某个接口。在拨号前缀正则表达式下，指定正则表达式。KInternet 中的拨号前缀（可由普通用户修改）必须符合此正则表达式。如果将此字段留空，用户则无法在不具备管理员权限的情况下设置其他拨号前缀。

在下一个对话框中，选择 ISP。要从您所在国家/地区的 ISP 的预定义列表中进行选择，请选择国家/地区。也可以单击新建打开一个对话框，从中为您的 ISP

提供数据。这些数据包括用于拨号连接的名称、ISP 的名称，以及 ISP 提供的登录名和密码。启用 *始终询问密码*，在您每次连接时都提示输入密码。

在最后一个对话框中，指定附加连接选项：

按需拨号

如果启用 *按需拨号*，请设置至少一个名称服务器。仅当您的因特网连接成本较低时才使用此功能，因为某些程序要求定期从因特网请求数据。

连接后修改 DNS

默认情况下启用此选项，其作用是在每次连接因特网时都更新名称服务器地址。

自动检索 DNS

如果提供者未在连接后传送其域名服务器，则禁用此选项并手动输入 DNS 数据。

自动重新连接

如果启用了此选项，则连接失败后会自动重新建立连接。

忽略提示

此选项将禁用对来自拨号服务器的任何提示检测。如果连接建立速度很慢，或根本不能建立连接，可尝试此选项。

外部防火墙接口

选择此选项将激活防火墙并将接口设置为外部的。这样您的系统就可以在连接到因特网期间防范外部攻击。

空闲超时（秒）

使用此选项可以指定网络不活动的时间，一超过该时间调制解调器即自动断开连接。

IP 详细信息

使用此选项可打开地址配置对话框。如果您的 ISP 没有为您的主机指派动态 IP 地址，请禁用 *动态 IP 地址*，然后输入主机的本地 IP 地址及远程 IP 地址。请向您的 ISP 询问这些信息。保持 *默认路由* 的启用状态，然后通过选择 *确定* 关闭该对话框。

选择下一步可返回初始对话框，其中显示调制解调器配置的概要。单击 *确定* 可关闭此对话框。

18.4.3 ISDN

使用此模块可以为系统配置一个或多个 ISDN 网卡。如果 YaST 没有检测到您的 ISDN 网卡，请在 *ISDN* 设备选项卡中单击添加，然后手动选择您的网卡。可以使用多个接口，但您可以为一个接口配置多个 ISP。在随后的对话框中，设置该网卡正常工作所需的 ISDN 选项。

图 18.5 ISDN 配置

 **contr0 的 ISDN 低级配置**
如果选择引导时，则在系统引导过程中 装载驱动程序。如果选择手动，则必须使用 rcisdn start 命令启动驱动程序。只有根用 ... [更多](#)

ISDN 卡信息

制造商

Abocom/Magitek

ISDN 卡

2BD1

驱动程序 (U):

HiSax driver

ISDN 协议

☒ Euro-ISDN (EDSSI)(E)

☐ 1TR6(G)

☐ 专线 (L)

☐ N11(I)

国家/地区 (C):

德国

代码 (D):

+49

区域号码 (A):

拨号前缀 (D):

☒ 启动 ISDN 日志 (O)

激活设备 (D):

在引导时

帮助

取消

后退

确定

在下一个对话框中（如图 18.5 “ISDN 配置” [213]所示），选择要使用的协议。默认值是 *Euro-ISDN (EDSSI)*，但是对于旧式或大型交换机，请选择 *1TR6*。如果是在美国，请选择 *N11*。在相关字段中选择您所在的国家/地区。相应的国家/地区代码将显示在该字段旁边的字段中。最后，提供您的区域号码和拨号前缀（如果需要）。如果不希望记录所有 ISDN 流量，则取消选中启动 *ISDN 日志* 选项。

激活设备定义应如何启动 ISDN 接口：使用在引导时可以在系统每次引导时初始化 ISDN 驱动程序。手动要求您以 root 的身份使用命令 rcisdn start 来装载 ISDN 驱动程序。热插拔，用于 PCMCIA 或 USB 设备，用于在插入设备后加载驱动程序。在完成这些设置后，请选择确定。

在下一个对话框中，为您的ISDN网卡指定接口类型，并将ISP添加到现有接口中。接口的类型可能是 SyncPPP 或 RawIP，但多数 ISP 以 SyncPPP 方式操作，如下文所述。

图 18.6 ISDN 接口配置



要为我的电话号码输入的值取决于特定的设置：

ISDN 网卡直接连接到电话插座

标准的 ISDN 线路提供三个电话号码（称为多用户号码或 MSN）。如果用户需要更多号码，最多可提供十个号码。必须在此处输入其中一个 MSN，但不要区号。如果输入的号码有误，您的电话运营商将自动退回到为您的 ISDN 线路指派的第一个 MSN。

ISDN 网卡连接到专用交换机

同样，配置可能随安装设备的不同而变化：

1. 适用于家庭的小型专用交换机 (PBX) 大多使用 Euro-ISDN (EDSS1) 协议进行内部呼叫。这些交换机具有内部 S0 总线，并对与它们连接的设备使用内部号码。

将其中一个内部号码用作您的 MSN。您应该至少能够使用支持直接向外拨号的交换机的 MSN 之一。如果无效，则尝试使用一个零。有关进一步的信息，请参见电话交换机随附的文档。

2. 为公司设计的大型电话交换机通常使用 1TR6 协议用于内部呼叫。它们的 MSN 称为 EAZ 并且通常对应直拨号码。要在 Linux 中配置，只需输入 EAZ 的最后一位即可。如果各种方法都行不通，可尝试 1 到 9 之间的各位数字。

要在下一个收费单位开始之前及时终止连接，请启用 *ChargeHUP*。但要记住，该选项不是对每个 ISP 都奏效。您也可以通过选择相应的选项启用信道绑定（多链接 PPP）。最后，您可以通过选择外部防火墙接口和重启防火墙为链接启用防火墙。要使不具备管理员权限的普通用户能够激活或停用接口，请选择通过 *Kinternet* 为非 root 用户启用设备控制。

单击细节打开一个对话框，在其中可实施更为复杂的连接方案，这与普通的家庭用户无关。通过选择确定退出细节对话框。

在下一个对话框中设置 IP 地址。如果您的提供者没有为您指定静态 IP，请选择动态 IP 地址。否则，根据 ISP 指定的信息，使用提供的字段输入您主机的本地 IP 地址及远程 IP 地址。如果接口应该成为与因特网连接的默认路由，请选择默认路由。每台主机都只能有一个接口配置为默认路由。选择下一步可退出此对话框。

使用随后的对话框，您可以设置您所在的国家/地区并选择 ISP。列表中的 ISP 都只是 call-by-call（通过呼叫进行呼叫）提供者。如果列表中未列出您的 ISP，请选择新建。随即打开提供者参数对话框，可以在其中输入 ISP 的所有详细信息。输入电话号码时，切勿在数字之间加空格或逗号。最后，输入 ISP 为您提供的登录名和密码。输完之后，请选择下一步。

要在独立工作站上使用按需拨号，还需指定名称服务器（DNS 服务器）。多数 ISP 都支持动态 DNS，这意味着每次用户连接时，都由 ISP 发送名称服务器的 IP 地址。不过，对于单个工作站，您仍然需要提供 192.168.22.99 之类的占位符地址。如果您的 ISP 不支持动态 DNS，请指定 ISP 的名称服务器 IP 地址。如果需要，可以为连接指定超时值 — 即网络不活动的时间（以秒计），一超过该时间即自动终止连接。选择下一步确认设置。YaST 将显示配置好的接口的概要。要激活这些设置，请选择确定。

18.4.4 电缆调制解调器

在某些国家/地区，人们往往通过有线电视网访问因特网。有线电视用户通常将调制解调器一端接在有线电视插座上，另一端与计算机网卡相连（使用 10Base-TG 双绞线）。随后电缆调制解调器就能通过固定 IP 地址提供专用因特网连接。

根据您的 ISP 提供的描述，配置网卡时选择 *动态地址* 或 *静态指派的 IP 地址*。目前多数提供商都使用 DHCP。通常只有特殊的公司帐户才使用静态 IP 地址。

有关配置电缆调制解调器的进一步信息，请联机访问 http://en.opensuse.org/SDB:Setting_Up_an_Internet_Connection_via_Cable_Modem_with_SuSE_Linux_8.0_or_Higher，阅读关于这一主题的“支持数据库”文章。

18.4.5 DSL

要配置 DSL 设备，请从 YaST 网络设备部分选择 *DSL* 模块。这个 YaST 模块包含若干对话框，可以在这些对话框中基于以下协议之一设置 DSL 链接参数。

- 以太网上的 PPP (PPPoE)
- ATM 上的 PPP (PPPoATM)
- 用于 ADSL 的 CAPI (Fritz 网卡)
- 点对点隧道协议 (Pptp) — 奥地利

在 *DSL 配置概述* 对话框的 *DSL 设备* 选项卡中，您将找到已安装 DSL 设备的列表。要更改 DSL 设备的配置，请在列表中选择该设备，然后单击 *编辑*。如果单击 *添加*，则可手动配置新 DSL 设备。

基于 PPPoE 或 PPTP 配置 DSL 连接时，要求已经正确设置相应的网卡。如果尚未这样做，应首先通过选择 *配置网卡* 来配置网卡（请参见第 18.4.1 节“使用 YaST 配置网卡” [201]）。使用 DSL 链接时，可以自动指派地址但并不通过 DHCP，这就是不应启用 *动态地址* 选项的原因。相反，应该为接口输入静态虚设地址，如 192.168.22.1。在 *子网掩码* 中，输入 255.255.255.0。如果配置的是独立工作站，则将 *默认网关* 留空。

提示


IP 地址字段和子网掩码中的值只是占位符。它们只用于初始化网卡，而不会将 DSL 链接表示成这样。

在第一个 DSL 配置对话框（请参见图 18.7 “DSL 配置” [217]）中，首先应选择 *PPP* 方式及 DSL 调制解调器连接到的以太网卡（多数情况下是 `eth0`）。然后

使用激活设备指定是否应在引导进程中建立 DSL 链接。单击通过 KInternet 为非 root 用户启用设备控制可授权普通用户无需根权限即可使用 KInternet 激活或停用接口。

在下一个对话框中，选择您的国家/地区并从该国家/地区运营的众多 ISP 中进行选择。随后的所有 DSL 配置对话框的详细信息都取决于目前已设置的选项，因此下面几段只对这些对话框进行了简要介绍。有关可用选项的详细信息，请阅读这些对话框中提供的详细帮助信息。

图 18.7 DSL 配置

**DSL 配置**

在此可设置 DSL 连接 最重要的设置。 首先，请选择 PPP 方式。可以是 PPP over Ethernet (← [更多](#))

DSL 连接设置

PPP 方式 (M):

PPP over Ethernet

PPP 方式相关设置

VPI/VCI (V):

Ethernet 网卡 (E)
未知设备
未知 - 未指派 IP 地址

更改设备 (C)

配置网卡 (C)


服务器名或 IP 地址 (S):


10.0.0.138


激活设备 (D):


手动

☒ 通过 KInternet 为非 root 用户启用设备控制 (N)

 帮助

 取消 (C)

 后退 (B)

 下一步 (N)

要在独立工作站上使用按需拨号，还需指定名称服务器（DNS 服务器）。多数 ISP 都支持动态 DNS — 每次用户连接时，都由 ISP 发送名称服务器的 IP 地址。不过，对于单个工作站，应提供 192.168.22.99 之类的占位符地址。如果您的 ISP 不支持动态 DNS，请输入 ISP 提供的名称服务器 IP 地址。

空闲超时（秒）定义网络不活动的时间，一超过该时间即自动终止连接。合理的超时值介于 60 到 300 秒之间。如果禁用了按需拨号，则最好将超时值设置为零以防止自动挂断。

T-DSL 的配置与 DSL 设置非常相似。只需将 *T-Online* 选为您的提供者，YaST 将打开 T-DSL 配置对话框。在此对话框中，提供 T-DSL 所需的一些其他信息——线路 ID、T-Online 号码、用户代码和密码。所有这些信息都会包含在订阅到 T-DSL 后收到的信息中。

18.5 NetworkManager

NetworkManager 是一种用于移动工作站的理想解决方案。有了 NetworkManager，您在移动中配置网络接口和切换网络时就无需担心了。NetworkManager 可以自动连接到已知的 WLAN 网络。如果有两种或多种连接选择，则连接速度可能会更快一些。

但是 NetworkManager 并非适用于所有情况的解决方案，因此您仍可在用于管理网络连接 (ifup) 的传统方法和 NetworkManager 之间进行选择。如果您希望使用 NetworkManager 管理网络连接，则在 YaST 网络设置模块中启用 NetworkManager（如第 22.2 节“启用 NetworkManager”[256]中所述），然后使用 NetworkManager 配置网络连接。有关使用案例的列表以及如何配置和使用 NetworkManager 的详细描述，请参考第 22 章 使用 NetworkManager [255]。

ifup 和 NetworkManager 之间的某些区别包括：

root 特权

如果使用 NetworkManager 进行网络安装，则可以使用一个小程序，随时从您的桌面环境内轻松地开关、停止或启动网络连接。NetworkManager 也可以改变和配置无线网卡连接，无需 root 特权。因此，NetworkManager 是一种用于移动工作站的理想解决方案。

使用 ifup 的传统配置也提供一些开关、停止或启动连接的方法，或需要或不需要用户参与（如用户管理设备），但通常需要 root 特权来改变或配置网络设备。这对于移动计算是个常见问题，因为移动计算不可能预配置所有的连接功能。

网络连接的类型

传统配置和 NetworkManager 都可以处理与无线网络（WEP、WPA-PSK 和 WPA-Enterprise 访问）、拨号连接以及使用 DHCP 和静态配置的有线网络之间的网络连接。它们也支持通过 VPN 的连接。

NetworkManager 尝试使用可用的最好连接使您的计算机随时保持连接状态。如果网络电缆意外断开，它将尝试重连接。它可以从您的无线连接列表中找到具有最佳信号强度的网络并自动用其进行连接。要用 ifup 获得同样的功能，需要花功夫进行配置。

18.6 手动配置网络连接

应该始终将手动配置网络软件作为最后的选择。建议使用 YaST。但是，对网络配置背景信息的了解将对您使用 YaST 有所帮助。

当内核检测到某个网卡并创建相应网络接口时，会根据设备发现的顺序或装载内核模块的顺序为设备指派一个名称。只有在控制程度极为简单或极为严格的硬件环境中，才能预测默认内核设备名称。允许在运行时添加或删除硬件的系统或支持自动配置设备的系统不能期望在各个重引导之间由内核指派稳定的网络设备名称。

但是，所有系统配置工具均依赖持久性接口名称。该问题通过 udev 解决。udev 永久网络生成器 (/etc/udev/rules.d/75-persistent-net-generator.rules) 会生成匹配硬件的规则（默认情况下使用其硬件地址），并为该硬件分配永久唯一接口。网络接口的 udev 数据库储存在文件 /etc/udev/rules.d/70-persistent-net.rules 中。文件中的每一行描述一个网络接口并指定其永久名称。系统管理员可通过编辑 NAME="" 项来更改指派的名称。也可以使用 YaST 修改永久规则。

表 18.5 “手动网络配置脚本” [219]总结了网络配置中涉及的最重要脚本。

表 18.5 手动网络配置脚本

命令	功能
if{up,down,status}	if* 脚本启动、停止网络接口或返回指定接口的状态。ifup 的手册页中提供了详细信息。

命令	功能
<code>rcnetwork</code>	<code>rcnetwork</code> 脚本可用于启动、停止或重新启动所有网络接口或某个特定接口。使用 <code>rcnetwork stop</code> 、 <code>rcnetwork start</code> 和 <code>rcnetwork restart</code> 可分别停止、启动和重新启动网络接口。如果要停止、启动或重新启动一个接口，请在命令后加上该接口名称，例如 <code>rcnetwork restart eth0</code> 。 <code>rcnetwork status</code> 命令显示接口的状态、其 IP 地址以及 DHCP 客户端是否正在运行。通过 <code>rcnetwork stop-all-dhcp-clients</code> 和 <code>rcnetwork restart-all-dhcp-clients</code> ，您可停止或重新启动运行在网络接口上的 DHCP 客户端。

有关 `udev` 和永久设备名的详细信息在[第 12 章 使用 `udev` 进行动态内核设备管理](#) [123]中有介绍。

18.6.1 配置文件

本节对网络配置文件进行了概述并解释了它们的作用和所使用的格式。

`/etc/sysconfig/network/ifcfg-*`

这些文件包含网络接口的配置。它们包含启动方式和 IP 地址等信息。可能的参数在 `ifup` 的手册页中有所介绍。另外，如果应将常规设置只用于一个接口，则文件 `dhcp`、`wireless` 和 `config` 中的所有变量都可用于 `ifcfg-*` 文件。

`/etc/sysconfig/network/{config, dhcp, wireless}`

文件 `config` 包含 `ifup`、`ifdown` 和 `ifstatus` 行为的常规设置。`dhcp` 包含用于无线 LAN 卡的 DHCP 和 `wireless` 设置。所有三个配置文件中的变量均已注释掉。`/etc/sysconfig/network/config` 中的一些变量也可用于 `ifcfg-*` 文件，在这些文件中它们被视为具有更高优先级。`/etc/sysconfig/network/ifcfg.template` 文件列出可以按接口指定的变量。但是，`/etc/`

sysconfig/network/config 中的多数变量是全局变量，不能在 ifcfg-files 中覆盖。例如，NETWORKMANAGER 或 NETCONFIG_* 变量是全局变量。

/etc/sysconfig/network/{routes,ifroute-*}

在这里确定 TCP/IP 包的静态路由。在 /etc/sysconfig/network/routes 文件中输入各种系统任务所需的所有静态路由：主机的路由、主机通过网关的路由以及网络的路由。对于需要个别路由的每个接口，定义另一个配置文件：/etc/sysconfig/network/ifroute-*。用接口名称替换 *。路由选择配置文件中的项如下所示：

# Destination	Dummy/Gateway	Netmask	Device
#			
127.0.0.0	0.0.0.0	255.255.255.0	lo
204.127.235.0	0.0.0.0	255.255.255.0	eth0
default	204.127.235.41	0.0.0.0	eth0
207.68.156.51	207.68.145.45	255.255.255.255	eth1
192.168.0.0	207.68.156.51	255.255.0.0	eth1

路由目标位于首列。此列可以包含网络或主机的 IP 地址，或者在有可访问名称服务器时，包含完全限定的网络或主机名。

第二列包含默认网关或通过其可访问主机或网络的网关。第三列包含网关后的网络或主机的子网掩码。例如，网关后主机的掩码为 255.255.255.255。

第四列只与本地主机连接的网络有关，如回路、以太网、ISDN、PPP 和虚拟设备。必须在此输入设备名。

（可选）可以使用第五列来指定路由的类型。不需要的列中应该包含一个减号 -，这样才能确保分析程序正确解析命令。关于详细信息，请参见 routes(5) 手册页。

/etc/resolv.conf

在此文件中指定主机所属的域（关键字 search）。同时列出的还有要访问的名称服务器地址的状态（关键字 nameserver）。可在文件中指定多个域名。当解析不是完全限定的名称时，将尝试通过附加单独的 search 项生成一个完全限定的名称。可在多行中指定多个名称服务器，每个以 nameserver 开始。注释以 # 符号开始。例 18.5 “/etc/resolv.conf” [222]显示 /etc/resolv.conf 的内容。

但是，`/etc/resolv.conf` 不应手动编辑。而是由 `netconfig` 脚本生成。要不使用 `YaST` 而定义静态 DNS 配置，请手动编辑 `/etc/sysconfig/network/config` 文件中的相应变量的值：`NETCONFIG_DNS_STATIC_SEARCHLIST`（用于主机名查询的 DNS 域名列表）、`NETCONFIG_DNS_STATIC_SERVERS`（用于主机名查询的名称服务器 IP 地址列表）和 `NETCONFIG_DNS_FORWARDER`（定义必须要配置的 DNS 转发器的名称）。要使用 `netconfig` 禁用 DNS 配置，请设置 `NETCONFIG_DNS_POLICY=''`。有关 `netconfig` 的更多信息，请参见 `man 8 netconfig`。

例 18.5 `/etc/resolv.conf`

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

`/sbin/netconfig`

`netconfig` 是一个用于管理附加网络配置设置的模块化工具。它合并了静态定义的设置和自动配置机制根据预定义策略以 `dhcp` 或 `ppp` 形式提供的设置。通过调用负责修改配置文件和重启动服务或相似操作的 `netconfig` 模块将所需更改应用于系统。

`netconfig` 识别三种主要操作。`netconfig modify` 和 `netconfig remove` 由诸如 `dhcp` 或 `ppp` 的守护程序在 `netconfig` 中提供或删除设置。仅 `netconfig update` 命令可用于用户：

`modify`

`netconfig modify` 命令修改特定于当前接口和服务的动态设置并更新网络配置。`Netconfig` 从标准输入或使用 `--lease-file filename` 选项指定的文件中读取设置，并将其储存在内部，直到系统重引导或者下一个修改或删除操作。已存在的相同接口和服务组合设置将会重写。该接口由 `-i interface_name` 参数指定。该服务由 `-s service_name` 参数指定。

`remove`

`netconfig remove` 命令为指定接口和服务组合删除由修改操作提供的动态设置并更新网络配置。该接口由 `-i interface_name` 参数指定。该服务由 `-s service_name` 参数指定。

update

`netconfig update` 命令使用当前设置更新网络配置。当策略或静态配置更改时，这非常有用。

`netconfig` 策略和静态配置设置可手动定义或者使用 YaST 在 `/etc/sysconfig/network/config` 文件中定义。由自动配置工具作为 `hdc` 或 `ppp` 提供的动态配置设置通过 `netconfig modify` 和 `netconfig remove` 操作由这些工具直接传递。`NetworkManager` 也使用 `netconfig modify` 和 `netconfig remove` 操作。启用 `NetworkManager` 时，`netconfig`（在策略模式 `auto` 中）仅使用 `NetworkManager` 设置，忽略任何其他接口使用传统 `ifup` 方法配置的设置。如果 `NetworkManager` 不提供任何设置，也将静态设置用作备份。不支持混合使用 `NetworkManager` 和传统 `ifup` 方法。

有关 `netconfig` 的更多信息，请参见 `man 8 netconfig`。

/etc/hosts

在此文件中，如 [例 18.6 “/etc/hosts” \[223\]](#) 中所示，将为主机名指派 IP 地址。如果未实施名称服务器，则将其建立 IP 连接的所有主机必须列在此处。在此文件中为每个主机输入一行，包含 IP 地址、完全限定的主机名和主机名。IP 地址必须在每行的开头，各项用空格和制表符隔开。注释总是以 `#` 符号开头。

例 18.6 */etc/hosts*

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
192.168.2.101 venus.example.com venus
```

/etc/networks

在这里，网络名称被转换为网络地址。格式类似于 `hosts` 文件的格式，只是网络名称在地址的前面。请参见 [例 18.7 “/etc/networks” \[223\]](#)。

例 18.7 */etc/networks*

```
loopback      127.0.0.0
localnet      192.168.0.0
```

/etc/host.conf

此文件控制名称解析，即通过解析程序库转换主机名和网络名称。此文件只用于链接到 libc4 或 libc5 的程序。对于当前的 glibc 程序，请参见 /etc/nsswitch.conf 中的设置。参数必须始终单独在一行上。注释以 # 符号开头。表 18.6 “/etc/host.conf 的参数” [224] 显示了可用的参数。例 18.8 “/etc/host.conf” [224]中显示了 /etc/host.conf 的示例。

表 18.6 /etc/host.conf 的参数

order hosts, bind	指定访问服务以进行名称解析的顺序。可用参数有（使用空格或逗号隔开）： hosts: 搜索 /etc/hosts 文件 bind: 访问名称服务器 nis: 使用 NIS
multi on/off	定义 /etc/hosts 中输入的主机是否可以具有多个 IP 地址。
nospoof on spoofalert on/off	这些参数影响名称服务器 spoofing，但对网络配置没有任何影响。
trim domainname	在主机名解析后，指定的域名与主机名分开（只要主机名包括域名）。此选项仅当来自本地域的名称在 /etc/hosts 文件中时才可用，但仍要用附加的域名进行组织。

例 18.8 /etc/host.conf

```
# We have named running
order hosts bind
# Allow multiple address
multi on
```

/etc/nsswitch.conf

GNU C Library 2.0 的引入与 名称服务转换 (NNS) 的引入是同时进行的。有关详细信息，请参见 `nsswitch.conf` (5) 手册页和 *GNU C Library 参见手册*。

查询的顺序是在文件 `/etc/nsswitch.conf` 中定义的。中显示了 `nsswitch.conf` 的示例。**例 18.9 “`/etc/nsswitch.conf`” [225]** 注释以 `#` 符号开头。在本例中，`hosts` 数据库下的项意味着通过 DNS 将请求发送到 `/etc/hosts` (files)。

例 18.9 `/etc/nsswitch.conf`

```
passwd:      compat
group:       compat

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files

netgroup:    files
automount:   files nis
```

表 18.7 “通过 `/etc/nsswitch.conf` 可用的数据库” [225] 中列出了 NSS 上可用的“数据库”。另外，近期将推出 `automount`、`bootparams`、`netmasks` 和 `publickey`。**表 18.8 “NSS“数据库”的配置选项” [226]** 中列出了 NSS 数据库的配置选项。

表 18.7 通过 `/etc/nsswitch.conf` 可用的数据库

<code>aliases</code>	<code>sendmail</code> 实施的邮件别名；请参见 <code>man5 aliases</code> 。
<code>ethers</code>	以太网地址。
<code>group</code>	对于 <code>getgrent</code> 使用的用户组。另请参见 <code>group</code> 的手册页。
<code>hosts</code>	<code>gethostbyname</code> 和类似函数使用的主机名和 IP 地址。

netgroup	网络中用于控制访问权限的主机和用户列表，请参见 netgroup(5) 手册页。
networks	getnetent使用的网络名称和地址。
password	getpwent 使用的用户密码；请参见 passwd(5) 手册页。
protocols	网络协议，由 getprotoent 使用；请参见 protocols(5) 手册页。
rpc	getrpcbyname和类似函数使用的远程过程调用名称和地址。
services	getservent使用的网络服务。
shadow	用户阴影密码，由 getspnam 使用；请参见 shadow(5) 手册页。

表 18.8 NSS“数据库”的配置选项

files	直接访问文件，例如 /etc/aliases
db	通过数据库访问
nis、nisplus	NIS，另请参见 第 4 章 <i>Using NIS</i> (↑安全指南)
dns	仅可用作 hosts 和 networks 的扩展名
compat	仅可用作 passwd、shadow 和 group 的扩展名

/etc/nscd.conf

此文件用于配置 nscd（名称服务缓存守护程序）。请参见 nscd(8) 和 nscd.conf(5) 手册页。默认情况下，passwd 和 groups 的系统项由 nscd 进行缓存。这对于目录服务（例如 NIS 和 LDAP）的性能很重要，因为如果不是这样，每次访问名称或组都需要网络连接。默认情况下，不对 hosts 进行缓

存，因为 `nscd` 中缓存主机的机制将导致本地系统无法信任正向和反向查找检查。请设置缓存 DNS 服务器，而不是让 `nscd` 缓存名称。

如果激活 `passwd` 的缓存，则通常需要 15 秒才能识别新添加的本地用户。通过使用命令 `rcnscdrestart` 重启动 `nscd` 可缩短此等待时间。

/etc/HOSTNAME

此文件提供不附带域名的主机名。当引导计算机时，此文件将被多个脚本读取。它可能只包含一行，该行中设置了主机名。

18.6.2 测试配置

向配置文件写配置之前，可对其进行测试。要设置测试配置，请使用 `ip` 命令。要测试连接，请使用 `ping` 命令。也可使用较早的配置工具 `ifconfig` 和 `route`。

命令 `ip`、`ifconfig` 和 `route` 会直接更改网络配置，而不会在配置文件中描述更改。如果未在正确的配置文件中输入配置，重引导时将丢失已更改的网络配置。

使用 ip 配置网络接口

`ip` 是用来显示和配置路由选择、网络设备、策略路由选择以及隧道的工具。它被设计为替换较早的工具 `ifconfig` 和 `route`。

`ip` 是非常复杂的工具。它的常用语法为 `ip options object command`。可使用以下对象：

链接

此对象表示网络设备。

地址

此对象表示设备的 IP 地址。

neighbour

此对象表示 ARP 或 NDISC 缓存项。

路由

此对象表示路由选择表项。

规则

此对象表示路由选择策略数据库中的规则。

maddress

此对象表示多路广播地址。

mroute

此对象表示多路广播路由缓存项。

tunnel

此对象表示 IP 上的隧道。

如果未提供命令，则将使用默认命令，通常为 `list`。

使用命令 `ip link set device_name command` 更改设备状态。例如，要取消激活设备 `eth0`，请输入 `ip link set eth0 down`。要重激活它，可使用 `ip link set eth0 up`。

激活设备后，可对设备进行配置。要设置 IP 地址，可使用 `ip addr add ip_address + dev device_name`。例如，要将接口 `eth0` 的地址设置为带标准广播（选项 `brd`）的 `192.168.12.154/30`，则输入 `ip addradd 192.168.12.154/30 brd + dev eth0`。

要拥有活动连接，还必须配置默认网关。要设置系统的网关，请输入 `ip route add gateway_ip_address`。要将一个 IP 地址转换为另一个 IP 地址，请使用 `nat:ip route add nat_ip_address via other_ip_address`。

要显示所有设备，可使用 `ip link ls`。要只显示正在运行的接口，可使用 `ip link ls up`。要打印设备的接口统计信息，可输入 `ip -s link ls device_name`。要查看设备的地址，请输入 `ip addr`。在 `ip addr` 的输出中，还可找到有关设备 MAC 地址的信息。要显示所有路由，可使用 `ip route show`。

有关使用 `ip` 的更多信息，请输入 `iphelp` 或查看 `ip(8)` 手册页。`help` 选项还可用于所有 `ip` 对象。例如，如果希望阅读 `ipaddr` 的帮助，请输入 `ipaddr`

help。可在 `/usr/share/doc/packages/iproute2/ip-cref.pdf` 中找到 `ip` 手册。

使用 `ping` 测试连接

`ping` 命令是用于测试 TCP/IP 连接是否有效的标准工具。它使用 ICMP 协议来将小数据包和 `ECHO_REQUEST` 数据报文发送到目标主机，并请求即时答复。如果发送有效，`ping` 将据此显示一条消息，指明网络链接基本有效。

`ping` 不仅能测试两台计算机之间的连接：它还能提供关于连接质量的基本信息。在例 18.10 “命令 `ping` 的输出” [229] 中，可查看 `ping` 输出示例。倒数第二行包含有关已发送的包数、丢失的包和 `ping` 运行的总时间量的信息。

您可以使用主机名或 IP 地址（例如 `pingexample.com` 或 `ping192.168.3.100`）作为目标。程序会一直发送包，直到您按 `Ctrl + C`。

如果只需要检查连接功能，则可使用 `-c` 选项来限制包数。例如，要将 `ping` 限制为三个包，请输入 `ping-c 3 example.com`。

例 18.10 命令 `ping` 的输出

```
ping -c 3 example.com
PING example.com (192.168.3.100) 56(84) bytes of data.
64 bytes from example.com (192.168.3.100): icmp_seq=1 ttl=49 time=188 ms
64 bytes from example.com (192.168.3.100): icmp_seq=2 ttl=49 time=184 ms
64 bytes from example.com (192.168.3.100): icmp_seq=3 ttl=49 time=183 ms
--- example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 183.417/185.447/188.259/2.052 ms
```

两个包之间的默认时间间隔为一秒。`ping` 提供了选项 `-i` 来更改时间间隔。例如，要将 `ping` 时间间隔延长为十秒，则输入 `ping-i 10 example.com`。

在带有多个网络设备的系统中，有时通过特定接口地址发送 `ping` 将会非常有用。要执行此操作，可将 `-I` 选项结合选定设备名称一起使用，例如 `ping-I wlan1 example.com`。

有关使用 `ping` 的更多选项和信息，请输入 `ping-h` 或查看 `ping (8)` 手册页。

使用 ifconfig 配置网络

ifconfig 是传统的网络配置工具。与 ip 相比，您只能将 ifconfig 用于接口配置。如果希望配置路由选择，可使用 route。

注意: ifconfig 和 ip

程序 ifconfig 已过时。请使用 ip。

毫无疑问，ifconfig 可显示当前活动接口的状态。在例 18.11 “命令 ifconfig 的输出”[230]中可看见 ifconfig 具有非常整齐和详细的输出。输出的第一行中还包含关于设备 MAC 地址的信息和 HWaddr 的值。

例 18.11 命令 ifconfig 的输出

```
eth0      Link encap:Ethernet  HWaddr 00:08:74:98:ED:51
          inet6 addr: fe80::208:74ff:fe98:ed51/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:634735 errors:0 dropped:0 overruns:4 frame:0
          TX packets:154779 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:1000
          RX bytes:162531992 (155.0 Mb)  TX bytes:49575995 (47.2 Mb)
          Interrupt:11 Base address:0xec80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8559 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8559 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:533234 (520.7 Kb)  TX bytes:533234 (520.7 Kb)

wlan1     Link encap:Ethernet  HWaddr 00:0E:2E:52:3B:1D
          inet addr:192.168.2.4  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20e:2eff:fe52:3b1d/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:50828 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43770 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:45978185 (43.8 Mb)  TX bytes:7526693 (7.1 MB)
```

有关使用 ifconfig 的更多选项和信息，请输入 ifconfig-h 或参见 ifconfig (8) 手册页。

使用 route 配置路由选择

route 是用于操作 IP 路由选择表的程序。可使用它来查看路由选择配置和添加或删除路由。

注意: route 和 ip

程序 route 已过时。请使用 ip。

如果需要有关路由选择配置的快速而又易懂的信息来确定路由选择问题，则route 将非常有用。要查看当前路由配置，请输入 route-n 作为 root。

例 18.12 命令 -n 的输出

```
route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags      MSS Window  irtt Iface
10.20.0.0        *               255.255.248.0    U          0 0        0 eth0
link-local       *               255.255.0.0      U          0 0        0 eth0
loopback         *               255.0.0.0        U          0 0        0 lo
default          styx.exam.com   0.0.0.0          UG         0 0        0 eth0
```

有关使用 route 的更多选项和信息，请输入 route-h 或参见 route (8) 手册页。

18.6.3 启动脚本

除了上面介绍的配置文件之外，还有多个脚本在引导计算机时装载网络程序。只要系统切换到某个多用户运行级别，就将启动这些脚本。中介绍了其中的一些脚本。表 18.9 “网络程序的一些启动脚本” [231]

表 18.9 网络程序的一些启动脚本

/etc/init.d/network	此脚本处理网络接口的配置。如果 network 服务未启动，则不实施任何网络接口。
/etc/init.d/xinetd	启动 xinetd。xinetd 可用于使服务器服务在系统上可用。例如，它可以在初始化 FTP 连接时启动 vsftpd。

<code>/etc/init.d/portmap</code>	启动 RPC 服务器所需的端口映射器，例如 NFS 服务器。
<code>/etc/init.d/nfsserver</code>	启动 NFS 服务器。
<code>/etc/init.d/postfix</code>	控制 postfix 进程。
<code>/etc/init.d/ypserv</code>	启动 NIS 服务器。
<code>/etc/init.d/ypbind</code>	

18.7 作为拨号助手的 smpppd

部分家庭用户不具备连接到因特网的专线。而是使用拨号连接。根据所用的拨号方法（ISDN 或 DSL），连接受 `ippd` 或 `pppd` 的控制。基本上，只要正确启动这些程序就可以联网了。

如果采用包月付费方式（拨号连接不产生任何附加费用），则只需启动相应的守护程序。用 KDE 小程序或命令行界面来控制拨号连接。如果因特网网关不是您所用的主机，最好通过网络主机来控制拨号连接。

这时就需要 `smpppd` 了。该程序为辅助程序提供统一的界面，并且可以双向执行。首先，它要对所需的 `pppd` 或 `ippd` 编程，并控制其拨号属性。然后，向用户程序提供各种提供商，并传送有关当前连接状态的信息。由于还可以通过网络来控制 `smpppd`，该程序适用于从专用子网中的工作站控制与因特网的拨号连接。

18.7.1 配置 smpppd

YaST 可以自动配置由 `smpppd` 提供的连接。同时还会预先配置实际的拨号程序 `KInternet` 和 `cinternet`。只有在配置 `smpppd` 的附加功能（如远程控制）时，才需要手动设置。

`smpppd` 的配置文件为 `/etc/smpppd.conf`。默认情况下并未启用远程控制。此配置文件最重要的选项包括：

`open-inet-socket = yes/no`

要通过网络控制 `smpppd`，必须将此选项设置为 `yes`。`smpppd` 的监听端口为 3185。如果此参数设置为 `yes`，则还需相应设置 `bind-address`、`host-range` 和 `password` 等参数。

`bind-address = ip address`

如果主机有多个 IP 地址，使用此参数可以确定 `smpppd` 应在哪个 IP 地址上接受连接。默认值是监听所有地址。

`host-range = min ipmax ip`

参数 `host-range` 用于定义网络范围。IP 地址属于这一范围的主机将被授予访问 `smpppd` 的权限。此范围之外的所有主机均不具备访问权。

`password = password`

通过指派密码可使客户机仅限于授权主机。由于这是个纯文本密码，不应高估该密码提供的安全性。如果未指派任何密码，所有客户机都有权访问 `smpppd`。

`slp-register = yes/no`

使用此参数，可以通过 SLP 在网络中声明 `smpppd` 服务。

关于 `smpppd` 的详细信息，请参见 `smpppd(8)` 和 `smpppd.conf(5)` 手册页。

18.7.2 配置供远程使用的 KInternet 和 cinternet

KInternet 和 cinternet 可用于控制本地或远程 `smpppd`。cinternet 是图形 KInternet 的命令行形式。要使这些实用程序可用于远程 `smpppd`，请手动编辑配置文件 `/etc/smpppd-c.conf` 或使用 KInternet。此文件仅使用四个选项：

`sites = list of sites`

此选项可以向前端通知 `smpppd` 的搜索位置。前端将按照在此指定的选项顺序来测试这些选项。`local` 选项规定建立到本地 `smpppd` 的连接。`gateway` 选项指向网关上的 `smpppd`。`config-file` 表示应建立到 `/etc/smpppd-c.conf` 文件中服务器和端口选项指定的 `smpppd` 的连接。`slp` 命令前端通过 SLP 连接到发现的 `smpppd`。

`server = server`

在此指定 `smpppd` 运行所在的主机。

`port = port`

在此指定 `smpppd` 运行所在的端口。

`password = password`

插入为 `smpppd` 选择的密码。

如果 `smpppd` 处于活动状态，现在即可访问它，例如通过 `cinternet--verbose --interface-list` 来访问。如果此时遇到困难，请参见 `smpppd-c.conf(5)` 和 `cinternet(8)` 手册页。

无线通讯

可以通过多种方法使用 Linux 系统与其他计算机、手提电话或外围设备进行通信。WLAN（无线 LAN）可用于将便携式计算机联网。蓝牙可用于将单独的系统部件（鼠标、键盘）、外围设备、手提电话、PDA 和单独的计算机互相连接。IrDA 通常用于与 PDA 或手提电话的通讯。全球移动通信系统 (UMTS) 也称为 3G，可以提供几种多媒体服务，例如浏览 Web 或收发消息。本章介绍这些技术及其配置。

19.1 无线 LAN

无线 LAN 已成为移动计算的不可缺少的一部分。当今，大多数笔记本电脑都配有内置 WLAN 卡。用于 WLAN 卡无线通讯的 802.11 标准是由 IEEE 组织制订的。最初，此标准实现的最大传送速率是 2 Mbit/s。此后，此标准进行了多次补充以提高数据传送速率。这些补充定义了调制、传送输出和传送速率等细节（请参见表 19.1 “各种 WLAN 标准的概述” [235]）。此外，许多公司实施了带专有或设计 (draft) 功能的硬件。

表 19.1 各种 WLAN 标准的概述

名称	频带 (GHz)	最大传送速率 (MBit/s)	记事
802.11 旧	2.4	2	已过时；目前市场上不销售采用此标准的最终设备
802.11a	5	54	不易受干扰

名称	频带 (GHz)	最大传送速率 (MBit/s)	记事
802.11b	2.4	11	较少使用
802.11g	2.4	54	广泛采用, 向后兼容 11b
802.11n draft	2.4 和/或 5	300	Common

SUSE® Linux Enterprise Desktop 不支持 802.11 旧卡。使用 802.11a、802.11b、802.11g 和 802.11n draft 的大多数卡受支持。新卡通常符合 802.11n draft 标准，但是使用 802.11g 的卡仍然可用。

19.1.1 功能

在无线联网中，会使用各种技术和配置来确保连接的快速、高质量和安全。不同的操作类型适合不同的设置。很难选择正确的身份验证方法。各种可用加密方法有各自的优点和缺陷。

无线网络基本上可分为受管网络和特殊网络。受管网络具有一个管理元素，即访问点。在这种方式（也称为基础结构方式）中，WLAN 工作站在网络中的所有连接都通过访问点运行，后者也可用作与以太网的连接。特殊网络没有访问点。各站之间直接通讯，因此专用网络通常比受管网络速度更快。但是，在专用网络中，传送范围和参与工作站的数目都受到很大限制。它们也不支持 WPA 身份验证。因此，通常使用接入点。甚至可以将 WLAN 卡用作访问点。某些卡支持此功能。

身份验证

与使用缆线连接的网络相比，无线网络中的数据更容易被截获，无线网络更容易受到攻击，所以各标准都包括了身份验证和加密方法。IEEE 802.11 标准最初的版本在术语 WEP 下对这些方法进行了描述。但是，WEP 被证明是不安全的（请参见“安全性”一节 [242]），因此 WLAN 行业（组织名为 Wi-Fi 联盟）制订了一个名为 WPA 的新扩展，用以弥补 WEP 的缺陷。后来的 IEEE 802.11i 标准（也称为 WPA2，因为 WPA 基于 802.11i 的草案版本）包括 WPA 和其他一些身份验证和加密方法。

为了确保只有经过授权的工作站才能连接，受管网络中使用了多种身份验证机制：

打开

开放系统是不要求身份验证的系统。任何工作站都可以加入网络。不过，可以使用 WEP 加密（请参见“**加密**”一节 [238]）。

共享密钥（按照 IEEE 802.11）

在此过程中，使用 WEP 密钥进行身份验证。但不建议采用此过程，因为它使 WEP 密钥容易受到攻击。攻击者所要做的一切就是侦听工作站和访问点之间的通讯足够长时间。在身份验证过程中，双方将交换相同的信息，一次使用的是加密形式，一次使用的是未加密形式。这使得可以使用适当的工具来重建密钥。由于方法使用 WEP 密钥来进行身份验证和加密，因此不能提高网络的安全性。具有正确 WEP 密钥的工作站可以进行身份验证、加密和解密。不具有密钥的工作站无法解密接收到的包。因此，无论它是否必须对本身进行身份验证都不能进行通讯。

WPA-PSK（按照 IEEE 802.1x）

WPA-PSK（PSK 代表“预共享密钥”）的工作方式与共享密钥过程类似。所有参与工作站和访问点需要相同的密钥。该密钥长度为 256 位，通常以密码短语形式输入。此系统不需要像 WPA-EAP 那样的复杂密钥管理，并且更适合个人使用。因此，有时将 WPA-PSK 称为 WPA“家庭”。“”

WPA-EAP（按照 IEEE 802.1x）

实际上，WPA-EAP 不是一个身份验证系统，而是一个传输身份验证信息的协议。WPA-EAP 用于保护企业中的无线网络。在个人网络中，很少使用 WPA-EAP。因此，WPA-EAP 有时称为 WPA“企业”。“”

WPA-EAP 需要 Radius 服务器来验证用户。EAP 提供了连接和身份验证服务器的三种不同方式：TLS (Transport Layer Security)、TTLS (Tunneled Transport Layer Security) 和 PEAP (Protected Extensible Authentication Protocol)。在 nutshell 中，这些选项的作用如下所示：

EAP-TLS

TLS 身份验证依赖于服务器和客户机的证书互相交换。首先，服务器为客户机（客户机会评估服务器）提供其证书。如果证书被认为有效，则接下来客户机会对服务器提供其证书。当 TLS 是安全的，它要求在网络中具有运转的认证管理基础结构。此基础结构在专用网络中很少见。

EAP-TTLS 和 PEAP

TTLS 和 PEAP 都是两个阶段的协议。在第一个阶段，将建立安全连接，在第二个阶段，将交换客户端身份验证数据。在需要认证管理的情况下，它们所需的认证管理费用比 TLS 要少得多。

加密

有多种加密方法可确保所有未授权用户不能读取无线网络中交换的数据包并且不能访问网络：

WEP（在 IEEE 802.11 中定义）

此标准使用 RC4 加密算法，最初密钥长度为 40 位，后来也使用 104 位的密钥。通常，将此长度声明为 64 位或 128 位，这取决于是否包括初始化矢量的 24 位。但是，此标准有一些缺陷。攻击者能够成功攻击此系统生成的密钥。不过，使用 WEP 总比根本不加密网络要好。

某些供应商实施了非标准的“动态 WEP”。它与 WEP 的工作完全相同，也具有相同弱点，不同之处在于密钥管理设备会定期更改密钥。

TKIP（在 WPA/IEEE 802.11i 中定义）

WPA 标准中定义的这一密钥管理协议使用与 WEP 相同的加密算法，但弥补了其缺陷。由于为每个数据包生成一个新密钥，从而有效阻止了对这些密钥的攻击。TKIP 与 WPA-PSK 一起使用。

CCMP（在 IEEE 802.11i 中定义）

CCMP 对密钥管理进行了描述。通常，它用于与 WPA-EAP 连接，但也可以与 WPA-PSK 一起使用。加密依照 AES 进行，该加密比 WEP 标准的 RC4 加密更强大。

19.1.2 用 YaST 配置

要配置无线网卡，请在 YaST 控制中心内选择 **网络设备 > 网络设置**。将打开“网络设置”对话框，可在其中配置打开常规网络设置。请参见 [第 18.4 节“使用 YaST 配置网络连接”](#) [201] 以获取有关常规网络配置的更多信息。系统检测到的所有网卡都在 **概述** 选项卡下列出。

从列表中选择无线网卡并单击 **编辑** 以打开“网卡设置”对话框。在 **地址** 选项卡下配置是使用动态 IP 地址，还是静态 IP 地址。也可以调整 **常规** 和 **硬件** 设置（如设

备激活或防火墙区域）和驱动程序设置。在大多数情况下，无需更改预配置的值。

单击下一步继续到无线网卡特定的配置对话框。如果正在使用 NetworkManager（请参见第 18.5 节“NetworkManager”[218]以获取更多信息），则无需调整无线设备设置，因为 NetworkManager 将按需设置这些设置 — 请继续单击下一步和是以完成配置。如果正在使用仅位于特定无线网络中的计算机，请在此处对 WLAN 操作进行基本设置。

图 19.1 YaST: 配置无线网卡



操作方式

我们可以将工作站以三种不同的方式集成到 WLAN 中。适用的模式取决于通讯所用的网络：特殊网络（对等网络，无访问点）、受管网络（网络由访问点管理）或者主网络（您的网卡应用作访问点）。要使用 WPA-PSK 或 WPA-EAP 方式，必须将操作方式设置为受管。

网络名称 (ESSID)

为实现相互通讯，无线网络中的所有工作站都需要相同的 ESSIDu163。如果未指定任何内容，则网卡会自动选择接入点，但它可能不是您所希望使用的。使用扫描网络以获取可用无线网络列表。

身份验证方式

为您的网络选择适合的身份验证方法：*不加密*、*WEP 开放式*、*WEP 共享密钥*、*WPA-EAP* 或 *WPA-PSK*。如果选择 *WPA 身份验证*，则必须设置网络名称 (ESSID)。

密钥输入类型

WEP 和 *WPA-PSK* 身份验证方法要求输入密钥。密钥必须输入为 *通行密码*、*ASCII 字符串* 或 *十六进制字符串*。

WEP 密钥

在此处输入默认密钥，或单击 *WEP 密钥* 进入高级密钥配置对话框。将密钥长度设置为 *128 位* 或 *64 位*。默认设置是 *128 位*。在对话框底部的列表区域中，最多可以指定 4 个不同的密钥，您的工作站将使用这些密钥进行加密。按 *设置默认密钥* 可将其中一个密钥定义为默认密钥。除非更改默认设置，否则 YaST 会将第一个输入的密钥用作默认密钥。如果删除了标准密钥，则必须将其他密钥中的一个手动标记为默认密钥。单击 *编辑* 可以修改现有列表项或创建新密钥。此时将出现一个弹出窗口，提示您选择输入类型（*通行密码*、*ASCII* 或 *十六进制*）。如果选择的是 *通行密码*，则输入一个单词或字符串，将从该单词或字符串按照先前指定的长度生成密钥。*ASCII* 要求为 *64 位* 密钥输入 5 个字符，为 *128 位* 密钥输入 13 个字符。如果选择的是 *十六进制*，则按照十六进制表示法为 *64 位* 密钥输入 10 个字符，或为 *128 位* 密钥输入 26 个字符。

WPA-PSK

要输入用于 *WPA-PSK* 的密钥，请选择输入方法 *通行密码* 或 *十六进制*。在 *通行密码* 方式下，输入必须为 8 到 63 个字符。在 *十六进制* 方式下，请输入 64 个字符。

专家设置

单击此按钮将打开一个对话框，用于对 *WLAN* 连接进行详细配置。通常无需更改预配置的设置。

通道

只有在 *特殊* 和 *主* 方式下才需要指定 *WLAN* 工作站要工作于的通道。在 *受控* 方式下，网卡将自动搜索访问点的可用通道。在 *专用* 方式下，可以选择提供的一个通道（11 到 14，具体取决于您所在国家/地区），用于在您的工作站和其他工作站之间进行通信。在 *主* 方式下，确定您的网卡应该在哪个通道上提供访问点功能。此选项的默认设置是 *自动*。

位速率

根据网络的性能，您可能要为从一点到另一点之间的传送设置特定位速率。在默认设置*自动*中，系统会尽可能地使用最高数据传送速率。一些 WLAN 卡不支持比特率设置。

接入点

在具有多个访问点的环境中，通过指定 MAC 地址可以预先选择多个访问点中的一个。

使用电源管理

当您在旅途中时，请使用省电技术来最大限度地延长电池的运行时间。有关电源管理的详细信息，请参考第 16 章 *电源管理* [163]。使用电源管理可能影响连接质量并增加网络延迟。

单击“下一步”完成设置。如果选择了 WPA-EAP 身份验证，需要首先执行另一个配置步骤，工作站才能在 WLAN 中部署。输入网络管理员提供的身份凭证。对于 TLS，请提供身份、客户机证书、客户机密钥和服务端证书。TTLS 和 PEAP 需要身份和密码。服务端证书和匿名身份为可选。YaST 在 /etc/cert 下搜索任何证书。因此，请将为您提供的证书保存在这个位置，并将对这些文件的访问限制为 0600（拥有者读写权限）。单击*细节*可进入 WPA-EAP 设置的高级身份验证对话框。选择 EAP-TTLS 或 EAP-PEAP 通信第二阶段的身份验证方法。如果在前面的对话框中已选择 TTLS，则选择任意、MD5、GTC、CHAP、PAP、MSCHAPv1 或 MSCHAPv2。如果已选择 PEAP，则选择任意、MD5、GTC 或 MSCHAPv2。如果自动确定的设置不起作用，则 PEAP 版本可用于强制使用特定的 PEAP 实施。

重要：无线网络中的安全性

确保使用所支持的身份验证和加密方法之一来保护您的网络通讯。如果未加密 WLAN 连接，则第三方便可以截获所有网络数据。即使进行弱加密 (WEP) 也比根本不加密要好。相关信息请参考“*加密*”一节 [238] 和“*安全性*”一节 [242]。

19.1.3 实用程序

包 wireless-tools 中包含可用于设置无线 LAN 特定参数和获取统计数字的实用程序。有关更多信息，请参见http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html。

19.1.4 建立 WLAN 的提示和技巧

这些提示可帮助精确调整 WLAN 的速度、稳定性和安全性。

稳定性和速度

无线网络的性能和可靠性主要取决于参与的工作站是否能够清楚地接收到来自其他工作站的信号。障碍物（例如，墙壁）极大地削弱了信号。信号强度越低，传送速率就越慢。在网络运行过程中，可以在命令行（Link Quality 字段）中使用 `Iwconfig` 实用程序或使用 `NetworkManager` 或 `KNetworkManager` 来检查信号强度。如果信号质量存在问题，可尝试将设备放在其他位置，或调整访问点天线的位置。很多 PCMCIA WLAN 卡都配有辅助天线，可充分提高接收效果。制造商指定的速率（例如 54Mbit/s）是一个额定值，它表示理论最大值。实际上，最大数据吞吐量不大于该值的一半。

安全性

如果要建立一个无线网络，则一定要记住，如果不实施任何安全措施，则传送范围内的任何人都可以方便地访问此网络。因此，一定要激活某种加密方法。所有 WLAN 卡和访问点都支持 WEP 加密。虽然这并非完全安全，但还是对潜在攻击者设置了一道屏障。WEP 通常可满足个人使用。WPA-PSK 的安全性更好，但不能在较早的访问点或具有 WLAN 功能的路由器中实施。在某些设备上，可以通过固件更新来实施 WPAu163 此外，尽管 Linux 在大多数硬件组件上支持 WPA，某些驱动程序仍然不提供 WPA 支持。如果 WPA 不可用，则使用 WEP 要好过不加密。在具有高级安全要求的企业中，无线网络工作时必须采用 WPA。

19.1.5 查错

如果 WLAN 卡未能作出响应，请检查您是否下载了所需的固件。请参见 `/usr/share/doc/packages/wireless-tools/README.firmware` 以获取更多信息。

多个网络设备

现在的便携式计算机通常都有网卡和 WLAN 卡，如果使用 DHCP（自动地址指派）来配置这两个设备，则您可能会遇到名称解析和默认网关的问题。可以 Ping 路由器但不能浏览因特网就是这方面问题的典型示例。位于 http://en.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients 的支持数据库提供了一篇有关这一主题的文章。

有关 Prism2 卡的问题

采用 Prism2 芯片的设备有多个驱动程序可用。不同的卡与不同的驱动程序之间的适用性是不一样的。使用这些卡时，只有在使用 hostap 驱动程序时，才能实施 WPA。如果这样的卡不能正常工作或根本不工作，或者您要使用 WPA，请参见 `/usr/share/doc/packages/wireless-tools/README.prism2`。

19.1.6 有关详细信息

Jean Tourrilhes 开发了用于 Linux 的无线工具，他的因特网网页上有很多关于无线网络的有用信息。请参见http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html。

网络中的 SLP 服务

制定 *服务位置协议* (SLP) 是为了简化本地网络中联网客户机的配置。要配置网络客户机（包括所有必需服务），管理员通常需要对网络中提供的服务器有详细了解。SLP 可以向本地网络中的所有客户机声明选中服务是否可用。支持 SLP 的应用程序则可以利用这一发布信息并进行自动配置。

SUSE® Linux Enterprise Desktop 支持使用 SLP 提供的安装源进行安装，并且包含许多集成了 SLP 支持的系统服务。YaST 和 Konqueror 都有适用于 SLP 的前端。您可以使用 SLP 为联网客户机（如系统上的安装服务器、文件服务器或打印服务器）提供核心功能。

重要: SUSE Linux Enterprise Desktop 中的 SLP 支持

提供 SLP 支持的服务包括 cupsd、rsyncd、ypserv、openldap2、ksysguardd、saned、kdm、vnc、login、smpppd、rpasswd、postfix 和 sshd（通过 fish）。

20.1 安装

默认情况下，仅安装 SLP 客户端和 slptool。如果希望通过 SLP 提供服务，则安装包 openslp-server。要安装包，请启动 YaST 并选择 **软件 > 软件管理**。立即选择 **过滤器 > 模式** 并单击 **其他服务器**。选择 openslp-server。确认已安装完成安装进程所需的包。

20.2 激活 SLP

要用 SLP 提供服务，您的系统上必须运行 `slpd`。如果计算机仅作为客户端操作，且不提供服务，则无需运行 `slpd`。类似 SUSE Linux Enterprise Desktop 中的大多数系统服务，`slpd` 守护程序通过单独的 `init` 脚本来控制。安装后，默认情况下停用守护程序。要将其临时激活，请作为 `root` 运行 `rcslpd start`，或运行 `rcslpd stop` 停止它。使用 `restart` 或 `status` 可分别执行重启或状态检查。如果希望引导时 `slpd` 始终处于活动状态，请在 YaST 系统 > 系统服务（运行级别）中启用 `slpd`，或以 `root` 身份运行 `insserv slpd` 命令。这包括要在引导时启动的一组服务中的 `slpd`。

20.3 SUSE Linux Enterprise Desktop 中的 SLP 前端

要查找网络中通过 SLP 提供的服务，请使用 SLP 前端。SUSE Linux Enterprise Desktop 包含几个前端：

slptool

`slptool` 是一个简单的命令程序，可用于在网络中发布 SLP 查询或专有服务。`slptool--help` 列出所有可用选项和功能。也可以从处理 SLP 信息的脚本调用 `slptool`。例如，要在当前网络中查找自行发布的所有网络时间服务器，请运行以下命令：

```
slptool findsrvs service:ntp
```

YaST

在 YaST 中，还有一个 SLP 浏览器。但是，该浏览器不能通过 YaST 控制中心使用。要启动此 YaST 模块，请作为 `root` 用户运行 `yast2 slp`。单击用户界面左侧的不同协议以了解关于相应服务的更多信息。

20.4 通过 SLP 提供服务

SUSE Linux Enterprise Desktop 中的许多应用程序都已使用 `libslp` 库集成了 SLP 支持。如果服务未符合 SLP 支持，请使用以下方法之一使其可通过 SLP 发布。

通过 `/etc/slp.reg.d` 进行的静态注册

为每个新服务创建单独的注册文件。下面显示了注册扫描仪服务的文件的示例：

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

此文件中最重要的一行是以 *service:* 开头的服务 URL。其中包含服务类型 (`scanner.sane`) 以及该服务在服务器上的地址。`$HOSTNAME` 自动用完整主机名替换。随后是可以找到相关服务的 TCP 端口的名称，端口与主机名之间用冒号分隔。然后输入服务的显示语言及以秒计的注册持续时间。应该用逗号分隔服务 URL 之后的各项内容。将注册持续时间设置为 0 到 65535 之间的值。0 表示禁止注册。65535 表示取消所有限制。

该注册文件还包含 `watch-port-tcp` 和 `description` 这两个变量。`watch-port-tcp` 通过使 `slpd` 检查相关服务的状态，链接 SLP 服务对该服务是否活动的发布。第二个变量为显示在适合的浏览器中的服务提供了更为准确的描述。

通过 `/etc/slp.reg` 进行的静态注册

此方法与使用 `/etc/slp.reg.d` 的步骤之间唯一的区别在于，所有服务在中心文件中分组。

使用 `slptool` 进行的动态注册

如果某个服务应动态注册，而无需配置文件，请使用 `slptool` 命令行实用程序。相同的实用程序还可用于取消某个现有服务产品的注册，而无需重新启动 `slpd`。

20.5 有关详细信息

以下来源提供了有关 SLP 的详细信息：

RFC 2608、2609、2610

RFC 2608 主要描述了 SLP 的定义。RFC 2609 更详细地描述了所用服务 URL 的语法；RFC 2610 则对通过 SLP 的 DHCP 进行了描述。

<http://www.openslp.org/>

OpenSLP 项目的主页。

`/usr/share/doc/packages/openslp`

此目录包含有关 SLP 的所有现有文档，其中包括 `README.SuSE`（包含 SUSE Linux Enterprise Desktop 细节、上述 RFC 和两个介绍性的 HTML 文档）。希望使用 SLP 功能的程序员可在 `openslp-devel` 包中包括的 *编程指南* 中找到更多信息。

使用 NTP 同步时间

NTP（网络时间协议）机制是用于同步网络上的系统时间的协议。首先，计算机从作为可靠时间源的服务器获得时间。然后将此计算机用作网络中其他计算机的时间源。这样做有双重目的：既可维护绝对时间，又可保持网络中所有计算机系统时间的同步。

维护确切的系统时间在许多情况下都非常重要。内置硬件(BIOS)时钟往往不能满足数据库这样的应用程序的要求。手动更正系统时间可能会导致许多严重问题，例如向后调整时间将使关键应用程序出现故障。在网络中，通常需要同步所有计算机上的系统时间，但是手动调整时间是一种不好的方法。`ntp`提供了解决这些问题的机制。该机制随时借助网络中的可靠时间服务器调整系统时间。它还支持对本地参考时钟（如无线电控制的时钟）进行管理。

注意

要通过 `Active Directory` 启用时间同步，请遵循 `Joining an AD Domain` (↑安全指南)中的描述。

21.1 使用 YaST 配置 NTP 客户机

`ntp`已预先设置为以本地计算机时钟为时间参考。但是，只有在没有更精确的时间源的情况下才使用(BIOS)时钟最为替代。`YaST`为NTP客户机的配置提供方便。对于不运行防火墙的系统，请使用快速或高级配置。对于受到防火墙保护的系统，高级配置可能打开 `SuSEfirewall2` 中需要的端口。

21.1.1 高级 NTP 客户机配置

您可手动或自动配置 NTP 客户端以通过 DHCP 获取网络中可用 NTP 服务器的列表。如果选择通过 *DHCP* 配置 NTP 守护程序，则无法使用以下所述的手动选项。

供客户端查询的服务器和其他时间资源列在 *常规设置* 选项卡的下半部分。使用 *添加*、*编辑* 和 *删除* 可按需修改此列表。 *显示日志* 使您能够查看客户机的日志文件。

单击 *添加* 可添加新的时间信息源。在随后的对话框中，选择要与其进行时间同步的源类型。下列选项可用：

服务器

另一个对话框使您可以选择 NTP 服务器。激活 *用于初始同步*，可以在引导系统时触发服务器和客户机之间的时间信息同步。 *选项* 使您可以指定 `ntpd` 的其他选项。

使用 *访问控制选项*，您可限制远程计算机通过您的计算机上运行的守护程序所能执行的操作。仅在选中安全设置选项卡上的 *限制 NTP 服务用于配置过的服务器* 后，才能启用此字段。选项对应于 `/etc/ntp.conf` 中的 `restrict` 子句。例如，`nomodify notrap noquery` 禁止服务器修改计算机的 NTP 设置并禁止使用 NTP 守护程序的陷阱工具（一种远程事件记录功能）。建议将这些限制用于超出您控制范围的服务器（例如在因特网上）。

有关详细信息，请参见 `/usr/share/doc/packages/ntp-doc`（`ntp-doc` 包的一部分）。

同级

同级是一台要与其建立对称关系的计算机：它将同时用作时间服务器和客户机。要在同一网络中用同级代替某个服务器，请输入系统的地址。该对话框的其他部分与 *服务器* 对话框相同。

无线电时钟

要在系统中使用无线电时钟来同步时间，请在此对话框中输入时钟类型、单元号码、设备名和其他选项。单击 *驱动程序校准* 可对该驱动程序进行微调。有关本地无线电时钟如何操作的详细信息，请参见 `/usr/share/doc/packages/ntp-doc/refclock.html`。

发出的广播

也可以通过在网络内广播的方式来传送时间信息和查询。在此对话框中，输入应将这类广播信息发送到的地址。除非使用了像无线电控制的时钟这样的可靠时间源，否则不要激活广播。

进来的广播

如果希望客户机通过广播接收信息，请在此字段中输入应接受来自哪个地址的相应数据包。

图 21.1 高级 NTP 配置：安全设置



在安全设置选项卡中，确定 ntpd 是否应在 chroot jail 中启动。默认情况下，在 *Chroot Jail* 中运行 NTP 守护程序是被激活的。在 chroot jail 中启动可以在遭受通过 ntpd 发起的攻击时提高安全性，因为这种方式可以防止攻击者危害整个系统。

限制 NTP 服务用于配置过的服务器通过禁止远程计算机查看和修改您计算机的 NTP 设置以及禁止使用用于远程事件记录的陷阱工具，从而增强了系统的安全

性。一旦启用，这些限制将适用于所有远程计算机，除非您在*常规设置选项卡*中针对时间源列表中的个别计算机覆盖了访问控制选项。对于所有其他远程计算机，仅允许查询本地时间。

如果 `SuSEfirewall2` 为默认的活动状态，则启用*打开防火墙中的端口*。如果保持端口的关闭状态，则不可能建立与事件服务器的连接。

21.2 在网络中手动配置 ntp

要使用网络中的时间服务器，最简便的方式就是设置服务器参数。例如，如果可以从网络访问名为 `ntp.example.com` 的时间服务器，请通过添加以下行将其名称添加到文件 `/etc/ntp.conf` 中：

```
server ntp.example.com
```

要添加更多时间服务器，请使用关键字 `server` 插入更多行。使用命令 `rcntpd start` 初始化 `ntpd` 后，等待时间稳定并且创建用于更正本地计算机时钟的偏移文件需要大约一个小时的时间。利用偏移文件，只要计算机一启动，就可以计算出硬件时钟的系统误差。可以立即使用更正功能，使系统时间保持较高的稳定性。

有两种方法可将 **NTP** 机制用作客户机：第一种方法是客户机可以定期从已知服务器查询时间。在存在许多客户机的情况下，这种方法会给服务器带来很高的负荷。第二种方法是客户机可以等待网络中的广播时间服务器发送 **NTP** 广播。这种方法的缺点在于服务器的可靠性是未知的，而且如果服务器发出错误信息将导致严重问题。

如果通过广播获取时间，则不需要服务器名称。此时只需在配置文件 `/etc/ntp.conf` 中输入 `broadcastclient` 一行。要以独占方式使用一个或多个已知时间服务器，请在以 `servers` 开头的行中输入它们的名称。

21.3 设置本地参考时钟

软件包 `ntp` 包含用于连接本地参考时钟的驱动程序。`ntp-doc` 包的文件 `/usr/share/doc/packages/ntp-doc/refclock.html` 中提供了受支持时钟的列表。每个驱动程序都有一个关联数字。在 `ntp` 中，实际配置是通过伪 IP 地址来完成的。时钟被输入 `/etc/ntp.conf` 文件，就像已经在网络中存在一样。

为此专门给它们指派了 `127.127.t.u` 格式的特殊 IP 地址。其中 *t* 代表时钟的类型并确定要使用的驱动程序，*u* 代表设备并确定要使用的接口。

通常，各个驱动程序都有特殊的参数来描述配置详细信息。文件 `/usr/share/doc/packages/ntp-doc/drivers/driverNN.html`（其中 *NN* 是驱动程序的编号）提供了有关特定类型时钟的信息。例如，“8 型”时钟（通过串行接口的无线电时钟）需要额外的方式更精确地指定时钟。以 Conrad DCF77 接收模块为例，该模块需要使用 `mode 5`。要使用此时钟作为首选参考，应指定关键字 `prefer`。由此构成的 Conrad DCF77 接收模块的完整 `server` 行如下：

```
server 127.127.8.0 mode 5 prefer
```

其他时钟也采用相同的模式。安装 `ntp-doc` 包之后，可以在目录 `/usr/share/doc/packages/ntp-doc/` 中找到 `ntp` 的文档。文件 `/usr/share/doc/packages/ntp-doc/refclock.html` 提供指向描述驱动程序参数的驱动程序页的链接。

使用 NetworkManager

NetworkManager 是用于便携式计算机和其他可移动计算机的理想解决方案。有了 NetworkManager，您在移动中配置网络接口以及切换有线或无线网络时就无需担心了。NetworkManager 可以自动连接到已知的无线网络。它还可并行管理多个网络连接，然后默认使用最快的连接。而且，您还可手动在可用网络之间切换，并使用系统盘中的小程序或控件管理网络连接。

在便携式计算机上，默认情况下 NetworkManager 处于活动状态。但是，可按照第 22.2 节“启用 NetworkManager”[256]中所述随时使用 YaST 将其激活或停用。

22.1 NetworkManager 的用例

NetworkManager 提供了完善而直观的用户界面，可使用户轻松地切换其网络环境。但是，NetworkManager 在以下情况下不适用：

- 您的计算机将为网络中的其他计算机（例如，DHCP 或 DNS 服务器）提供网络服务。
- 计算机为 Xen 服务器或系统是 Xen 内的虚拟系统时。
- 您想在网络配置管理中使用 SCPM。要同时使用 SCPM 和 NetworkManager，请在 SCPM 配置中禁用网络资源。

22.2 启用 NetworkManager

如果您希望使用 NetworkManager 管理网络连接，则在 YaST 网络设置模块中启用 NetworkManager。要启用 NetworkManager，请按如下操作：

- 1 运行 YaST 然后转到 *网络设备 > 网络设置*
- 2 将打开 *网络设置* 对话框。转到 *全局选项* 选项卡。
- 3 在 *网络设置方法* 字段中，激活通过 *NetworkManager* 的用户控制方法。
- 4 单击 *完成*。
- 5 选择用于管理网络连接的方法后，请使用通过 DHCP 或静态 IP 地址的自动配置安装网卡或配置您的调制解调器（对于拨号连接，请使用 *网络设备 > 调制解调器*）。要配置内部或 USB ISDN 调制解调器，请选择 *网络设备 > ISDN*。要配置内部或 USB DSL 调制解调器，请选择 *网络设备 > DSL*。

在 [第 18.4 节“使用 YaST 配置网络连接”](#) [201] 和 [第 19.1 节“无线 LAN”](#) [235] 中查找使用 YaST 进行网络配置的详细描述。

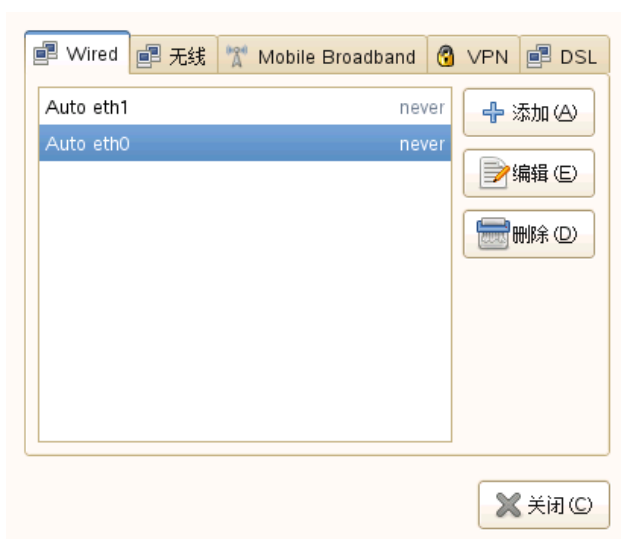
启用 NetworkManager 后，按照 [第 22.3 节“配置网络连接”](#) [256] 中所述使用 NetworkManager 配置网络连接。

如果希望停用 NetworkManager 并以传统方法控制网络，请在 *网络设置方法* 字段中选择通过 *ifup* 的传统方法选项。

22.3 配置网络连接

在 YaST 中启用 NetworkManager 后，从 GNOME 控制中心或 KDE 4 中的“个人设置”提供的对话框中配置网络连接。如果使用 GNOME，则从主菜单中启动 GNOME 控制中心，然后选择 *系统 > 网络配置* 以打开 *网络配置* 对话框。如果使用 KDE，则通过单击 *配置桌面* 从主菜单中启动个人设置，然后选择 *高级 > 网络设置* 以打开 *网络设置* 对话框。

图 22.1 GNOME 网络配置对话框



或者，您通过单击 *配置 (KDE)* 或右键单击 GNOME 小程序然后选择 *编辑连接*，从而在系统盘的 NetworkManager 小程序/控件中启动配置对话框。

GNOME 和 KDE 4 配置对话框显示针对所有网络连接类型的选项卡，如有线、无线、移动宽带、DSL 和 VPN 连接。NetworkManager 还支持连接到受保护的 802.1X 网络。

要添加新连接，请单击要使用的连接类型的选项卡，然后单击 *添加*。输入 *连接名称* 和连接细节。如果每个连接类型有多个物理设备（例如计算机装有两个以太网卡或两个无线网卡），请指定该设备的 *MAC 地址*（硬件地址），以便绑定到该设备的连接。单击 *确定* 或 *应用* 确认您的设置。新配置的网络连接现在显示在单击 NetworkManager 小程序或控件可获取的可用网络列表中。

注意：隐藏网络

要连接到“隐藏”网络（没有广播其服务的网络），您必须知道该网络的扩展服务集标识符 (ESSID)，因为无法自动检测该网络。在这种情况下，请输入 ESSID 和加密参数（如果需要）。

编辑每个连接时，您还可定义 NetworkManager 是自动使用此连接（激活 *自动连接*）还是在系统范围内使用此连接（激活 *对所有用户可用*）。此类系统连接可

以由所有用户共享，并在启动NetworkManager之后（任何用户登录之前）立即可用。要创建和编辑系统连接，需要 `root` 权限。

22.4 使用 KDE NetworkManager 控件

在 KDE 4 中，用于控制 NetworkManager 的 KNetworkManager 小程序已替换为 NetworkManager 控件。控件是指可以集成到您的桌面或面板的小型应用程序。如果已经为 NetworkManager 控件设置了网络，则控件通常通过桌面环境自动启动且显示为系统托盘中的图标。

NetworkManager 控件将当前网络状态显示为一个图标，并使用通知报告更改。使用该控件可配置新网络连接、手动选择其他网络连接、禁用无线网络或切换到脱机模式（如果需要）。图标的外观取决于当前网络连接的类型和状态。将鼠标光标置于图标上方可查看有关连接的细节。

NetworkManager 区分两种类型的连接，即可信和不可信。可信连接是您明确选择过的任何网络。所有其他连接均为不可信连接。右键单击连接图标可显示您在过去至少使用过一次的连接的列表。当前使用的连接已在菜单中标记。

左键单击任何连接小程序可随时选择其他网络连接。这种选择优先于自动选择网络。所选网络只要存在尽可使用，这意味着插入网络数据线不会自动切换到有线网络连接。

22.5 使用 GNOME NetworkManager 小程序

在 GNOME 中，可通过 GNOME NetworkManager 小程序控制 NetworkManager。如果为 NetworkManager 控件设置了网络，则小程序通常通过桌面环境自动启动且显示为系统盘中的图标。

如果您的系统盘没有显示 GNOME NetworkManager 小程序，该小程序可能未启动。按 `Alt + F2` 键并输入 `nm-applet` 可手动启动它。

22.5.1 连接到有线网络

如果您的计算机已通过网络电缆连接到现有网络上，则使用 NetworkManager 小程序选择网络连接。

- 1 单击小程序的图标可显示具有可用网络的菜单。在菜单中选择当前使用的连接。
- 2 要切换到另一个网络，请从列表中选择它。
- 3 要关闭所有网络连接，包括有线的和无线的，请右键单击小程序图标并取消选中 *启用联网*。

22.5.2 连接到无线网络

可用的可见无线网络在无线网络下的 GNOME NetworkManager 小程序菜单中列出。每个网络的信号强度也会显示在菜单中。加密无线网络是用保护物图标标记的。

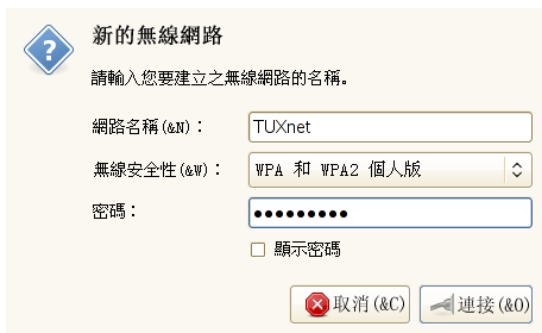
过程 22.1 连接到无线网络

- 1 要连接到无线网络，请单击小程序图标并从可用无线网络中选择一项。
- 2 如果网络已加密，将打开一个对话框。选择网络使用的无线安全类型并输入相应的密码。
- 3 要连接到没有广播其服务集标识符 (ESSID) 并因此无法自动检测的网络，请单击 NetworkManager 图标并选择 *连接到其他无线网络*。
- 4 在打开的对话框中，输入 ESSID 并在需要时设置加密参数。
- 5 要禁用无线联网，请右键单击小程序图标并取消选中 *启用无线*。这对于飞机上或其他不允许无线联网的环境是非常有用的。

22.5.3 将无线网卡配置为接入点

如果无线网卡支持接入点方式，则可以使用 NetworkManager 来进行配置。

1 单击创建新的无线网络。



2 添加网络名称并在无线安全性对话框中设置加密。

重要: 不受保护的无线网络是有安全风险的。

如果将无线安全设置为无，任何人都可以连接到您的网络，重复使用您的连接并截获您的网络连接。要限制对访问点的访问，确保连接安全，请使用加密。您可在基于 WEP 和 WPA 的各种加密方法之间进行选择。如果不能肯定哪种技术最适合，请阅读[“身份验证”一节](#) [236]。

22.6 NetworkManager 和 VPN

NetworkManager 支持多种虚拟专用网 (VPN) 技术：

- NovellVPN—包 NetworkManager-novellvpn
- OpenVPN—包 NetworkManager-openvpn
- vpnc (Cisco) — 包 NetworkManager-vpnc
- PPTP (点对点隧道协议) — 包 NetworkManager-pptp

要将 VPN 用于 NetworkManager，请首先安装相应的 VPN 包。对于每种 VPN 技术，您需要两个包：其中一个即为上述的包（提供对 NetworkManager 的常规支持），另一个是用于您的小程序且特定于桌面的相应包。

对于 KDE，请选择以下包之一：

- 对 KNetworkManager 的 NovellVPN 支持 — 包
`NetworkManager-novellvpn-kde4`
- 对 KNetworkManager 的 OpenVPN 支持 — 包
`NetworkManager-openvpn-kde4`
- 对 KNetworkManager 的 vpnc (Cisco) 支持 — 包
`NetworkManager-vpnc-kde4`

对 KDE 的 PPTP 支持目前尚不可用，但正在努力解决。

对于 GNOME，请选择以下包之一：

- 对 GNOME NetworkManager 小程序的 NovellVPN 支持 — 包
`NetworkManager-novellvpn-gnome`
- 对 GNOME NetworkManager 小程序的 OpenVPN 支持 — 包
`NetworkManager-openvpn-gnome`
- 对 GNOME NetworkManager 小程序的 vpnc (Cisco) 支持 — 包
`NetworkManager-vpnc-gnome`
- 对 GNOME NetworkManager 小程序的 PPTP（点对点隧道协议）支持 — 包
`NetworkManager-pptp-gnome`

在安装包后，按照第 22.3 节“配置网络连接”[256]中所述配置 VPN 连接。

22.7 NetworkManager 和安全性

NetworkManager 区分两种类型的无线连接，即可信和不可信。可信连接是您过去明确选择的任何网络。所有其他连接均为不可信连接。可信连接用访问点的名称和 MAC 地址识别。使用 MAC 地址可以确保带有可信连接名称的不同访问点不可使用。

NetworkManager 定期扫描是否存在可用的无线网络。如果找到多个可信网络，则自动选择最近使用的可信网络。如果所有网络均不可信，则 NetworkManager 等待您的选择。

如果加密设置改变，但名称和 MAC 地址不变，则 NetworkManager 将尝试连接，但首先会要求您确认新的加密设置并提供任意更新（如新密钥）。

NetworkManager 可识别两种类型的连接：用户和系统连接。用户连接是第一个用户登录时对 NetworkManager 可用的连接。会向该用户询问任何必需的身份凭证，当该用户注销时，连接会断开并从 NetworkManager 中删除。定义为系统连接的连接可以由所有用户共享，并在启动 NetworkManager 之后（任何用户登录之前）立即可用。如果是系统连接，必须在创建连接时提供所有身份凭证。此类系统连接可用于自动连接到要求授权的网络。有关如何配置用户连接或系统连接的信息，请参见第 22.3 节“配置网络连接”[256]。

如果从使用无线连接切换到脱机模式，NetworkManager 将使 ESSID 显示为空白。这可以确保断开网卡连接。

22.7.1 储存密码和身份凭证

如果不想每次连接到加密网络时都要再次输入身份凭证，则可以使用特定于桌面的工具 GNOME 密钥环管理器或 KWalletManager 将身份凭证加密储存在磁盘上，并用主密码保护。有关 GNOME 密钥环管理器的详细信息，请参考第 2.4.3 节“Managing Keyrings”（第 2 章 *Customizing Your Settings*, ↑GNOME 用户指南）。

NetworkManager 还从证书储存检索用于安全连接（例如加密的有线、无线或 VPN 连接）的证书。有关更多信息，请参考第 13 章 *Certificate Store* (↑安全指南)。

另一个选项是在 Novell CASA 中使用一次签到。一次签到是一种访问控制方法，它让用户身份验证一次，即可以访问多个软件系统的资源。如果系统中配置了 Novell CASA，NetworkManager 将不会请求其他密码来解除 GNOME 密钥环管理器锁定。相反，在用户登录到桌面时，密钥环将自动解除锁定。有关 Novell CASA 的详细信息，请参考<http://developer.novell.com/wiki/index.php/Special:Downloads/casa>。

22.8 常见问题

下面是关于使用 NetworkManager 配置特殊网络选项的一些常见问题。

如何将连接绑定到特定设备？

默认情况下，NetworkManager 中的连接是特定于连接类型的：它们适用于同一类型的所有物理设备。如果每个连接类型有多个物理设备（例如，计算机装有两个以太网卡），您可以将通过显式指定该设备的硬件地址（或 MAC 地址）而将连接绑定到特定设备。

在连接信息（在小程序/控件中提供）中查找设备的 MAC 地址，或使用诸如 nm-tool 或 ifconfig 等命令行工具的输出。然后从 GNOME 控制中心通过系统 > 网络配置，或在 KDE 4 中从个人设置的高级 > 网络设置启动用于配置网络连接的对话框。选择要修改的连接，然后单击编辑。在有线或无线选项卡上，输入设备的 MAC 地址，并通过单击确定确认更改。

如果检测到同一 ESSID 有多个接入点，如何指定特定接入点？

当有不同无线波段 (a/b/g/n) 的多个接入点时，默认情况下会自动选择信号最强的接入点。要覆盖此值，配置无线连接时请使用 BSSID 字段。

基本服务集标识 (BSSID) 可唯一标识每个基本服务集。在基础结构基本服务集中，BSSID 是无线接入点的 MAC 地址。在独立（特别）基本服务集中，BSSID 是本地管理的 MAC 地址（从 46 位数字随机生成）。

从 GNOME 控制中心通过系统 > 网络配置，或在 KDE 4 中从个人设置的高级 > 网络设置启动用于配置网络连接的对话框。选择要修改的无线连接，然后单击编辑。在无线选项卡上，输入 BSSID。

如何将网络连接与其他计算机共享？

主设备（连接到因特网的设备）不需要任何特殊配置。但是，需要如下配置连接到本地集线器或计算机的设备：

1. 从 GNOME 控制中心通过系统 > 网络配置，或在 KDE 4 中从个人设置的高级 > 网络设置启动用于配置网络连接的对话框。选择要修改的连接，然后单击编辑。切换至 IPv4 设置选项卡。从方法下拉列表中，选择与其他计算机共享。这将启用 IP 通讯转发并运行该设备上的 DHCP 服务器。在 NetworkManager 中确认更改。
2. 由于 DHCP 服务器使用端口 67，请确保该端口没有被防火墙阻止：在共享连接的计算机上，启动 YaST 并选择安全和用户 > 防火墙。切换到允许的服务类别。如果 DHCP 服务器尚未显示为允许的服务，请从待允许的服务中选择 DHCP 服务器，然后单击添加。在 YaST 中确认更改。

如何对自动（DHCP、PPP、VPN）地址提供静态 DNS 信息？

如果 DHCP 服务器提供无效的 DNS 信息（和/或路由），则可以覆盖它。从 GNOME 控制中心通过系统 > 网络配置，或在 KDE 4 中从个人设置的高级 > 网络设置启动用于配置网络连接的对话框。选择要修改的连接，然后单击编辑。切换到 IPv4 设置选项卡，从方法下拉列表中，选择仅自动 (DHCP) 地址。在 DNS 服务器和搜索域字段中输入 DNS 信息。单击路由添加更多路由或覆盖自动路由。确认更改。

有用户登录密码保护的网路前，如何使 NetworkManager 连接到该网络？

定义可以用于此类用途的系统连接。有关更多信息，请参考第 22.7 节“NetworkManager 和安全性”[261]。

22.9 查错

可能出现连接问题。有关 NetworkManager 的一些常见问题包括小程序未启动、VPN 选项缺失以及 SCPM 问题。解决方法和预防这些问题的方法随使用的工具而定。

NetworkManager 桌面小程序/控件未启动

如果为 NetworkManager 控件设置了网络，则 GNOME NetworkManager 小程序或 KDE NetworkManager 控件应自动启动。若小程序/控件未启动，则按照第 22.2 节“启用 NetworkManager”[256]中所述检查是否在 YaST 中启用了 NetworkManager。然后确保也已安装用于您的桌面环境的相应包。如果正在使用 KDE 4，则包为 NetworkManager-kde4。对于 GNOME 用户，包为 NetworkManager-gnome。

如果 GNOME 桌面小程序已安装，但出于某种原因没有运行（可能您意外退出），则使用 nm-applet 命令手动启动。

如果您的 KDE 4 系统盘未显示网络连接的任何图标（在 YaST 中从静态网络配置切换到使用 NetworkManager 的用户控制后可能出现的情况），请将 NetworkManager 控件添加到面板：右键单击面板上的空增补程序，然后选择面板选项 > 添加控件。（如果您的桌面对象当前已锁定，则可能首先需要单击解除控件锁定，然后才能添加任何对象。）在显示的对话框中，选择 NetworkManager，然后单击添加控件。

NetworkManager 小程序/控件不包括 VPN 选项

对 NetworkManager、小程序以及针对 NetworkManager 的 VPN 的支持在不同的包中分发。如果 NetworkManager 小程序/控件不包括 VPN 选项，请检查带有 NetworkManager 的 VPN 支持的包是否已安装。有关详细信息，请参见第 22.6 节“NetworkManager 和 VPN” [260]。

SCPM 不切换网络配置

您可能正在将 SCPM 与 NetworkManager 一起使用。NetworkManager 当前无法使用 SCPM 配置文件。当 SCPM 配置文件也更改了网络设置时，请勿将 NetworkManager 与 SCPM 一起使用。要同时使用 SCPM 和 NetworkManager，请在 SCPM 配置中禁用网络资源。

没有可用的网络连接

如果您已正确配置网络连接并且网络连接的所有其他组件（路由器等等）也已启动并在正常运行，则重新启动计算机上的网络接口有时可能有帮助。要执行此操作，请作为 root 登录到命令行，然后运行 `rcnetwork restart`。

22.10 更多信息

可在以下网站和目录中找到有关 NetworkManager 的更多信息：

- <http://www.gnome.org/projects/NetworkManager/>—NetworkManager 项目页
- 有关 KDE NetworkManager 控件的更多信息，请参考 <http://userbase.kde.org/KNetworkManager>。
- 另请检查以下目录中的信息是否包含有关 NetworkManager 和 GNOME NetworkManager 小程序及 KDE NetworkManager 控件的最新信息：`/usr/share/doc/packages/NetworkManager/`、`/usr/share/doc/packages/NetworkManager-kde4/` 和 `/usr/share/doc/packages/NetworkManager-gnome/`

Samba

使用 Samba，可以将 Unix 计算机配置为 Mac OS X、Windows 和 OS/2 计算机的文件和打印服务器。Samba 已经发展成为一个功能完备且相当复杂的产品。使用 YaST、SWAT（Web 界面）或通过手动编辑配置文件来配置 Samba。

23.1 术语

以下是 Samba 文档和 YaST 模块中使用的一些术语。

SMB 协议

Samba 使用基于 NetBIOS 服务的 SMB（服务器消息块）协议。Microsoft 发布该协议以便其他软件制造商能够与 Microsoft 域网络建立连接。使用 Samba 时，SMB 协议在 TCP/IP 协议之上工作，所以必须在所有客户机上安装 TCP/IP 协议。

CIFS 协议

（常用因特网文件系统）协议是 Samba 支持的另一种协议。CIFS 定义网络中使用的标准远程文件系统访问协议，使用户组能够一起工作并在网络中共享文档。

NetBIOS

NetBIOS 是为用于提供名称服务的计算机之间进行通讯而设计的软件接口 (API)。它使连接到网络的计算机能够为自己保留名称。之后便可以根据名称对这些计算机进行寻址。没有任何中心进程来检查这些名称。网络上的任何计算机均可以保留所需数量的名称，前提是这些名称均未使用。可以为不同的网络体系结构实施 NetBIOS 接口。NetBEUI 是与网络硬件结合相

对密切的一种实施，但它常被称为 NetBIOS。使用 NetBIOS 实施的网络协议包括 Novell 的 IPX（通过 TCP/IP 的 NetBIOS）和 TCP/IP。

通过 TCP/IP 发送的 NetBIOS 名称与 `/etc/hosts` 中使用的名称或 DNS 定义的名称没有相同之处。NetBIOS 使用它自己的、完全独立的命名约定。但为了方便管理，仍建议您使用与 DNS 主机名对应的名称，或本机使用 DNS。Samba 默认采用这种方式。

Samba 服务器

Samba 服务器向客户端提供 SMB/CIFS 服务和 NetBIOS over IP 命名服务。对于 Linux，Samba 服务器有三个守护程序：smnd 用于 SMB/CIFS 服务，nmbd 用于命名服务，winbind 用于身份验证。

Samba 客户机

Samba 客户机是一种能够通过 SMB 协议从 Samba 服务器使用 Samba 服务的系统。所有常见操作系统（Mac OS X、Windows 和 OS/2 等）都支持 SMB 协议。必须在所有计算机上安装 TCP/IP 协议。Samba 为多种不同的 UNIX 系统提供客户机。对于 Linux，有一个用于 SMB 的内核模块，它允许在 Linux 系统级别上集成 SMB 资源。不需要对 Samba 客户机运行任何守护程序。

共享

SMB 服务器通过共享为其客户端提供资源。共享就是服务器上的打印机和目录及其子目录。可以通过名称来导出并访问共享。可以将共享名称设置为任何名称 — 它不一定是导出目录的名称。也可以为打印机指派一个名称。客户机可以根据打印机的名称来访问打印机。

DC

域控制器 (DC) 是处理域中的帐户的服务器。为了复制数据，一个域中可有更多域控制器可用。

23.2 配置 Samba 服务器

关于配置 Samba 服务器，请参见 SUSE Linux Enterprise Server 文档。

23.3 配置客户机

客户机只能通过 TCP/IP 访问 Samba 服务器。NetBEUI 和通过 IPX 的 NetBIOS 不能与 Samba 共用。

23.3.1 使用 YaST 配置 Samba 客户机

配置 Samba 客户机来访问 Samba 服务器上的资源（文件或打印机）。在 *网络服务 > Windows 域成员资格* 对话框中输入域或工作组。如果激活将 *SMB* 信息也用于 *Linux* 身份验证，则用户身份验证将在 Samba 服务器上运行。在完成所有设置后，单击完成完成配置。

23.4 将 Samba 用作登录服务器

在主要由 Windows 客户机组成的网络中，使用户只能使用有效帐户和密码进行注册通常是最好的选择。在基于 Windows 的网络中，此任务由主域控制器 (PDC) 来处理。您可以使用配置为 PDC 的 Windows NT 服务器，但是此任务也可以借助 Samba 服务器来完成。中显示了必须在 `smb.conf` 的 `[global]` 部分设置的项。[例 23.1 “smb.conf 中的 global 部分” \[269\]](#)

例 23.1 `smb.conf` 中的 `global` 部分

```
[global]
    workgroup = TUX-NET
    domain logons = Yes
    domain master = Yes
```

如果使用加密密码进行校验，则 Samba 服务器必须能够处理加密密码。

`[global]` 部分中的 `encrypt passwords = yes` 项启用了此功能（对于 Samba 版本 3，这是默认设置）。此外，还需要以适合 Windows 的加密格式来准备用户帐户和密码。使用命令 `smbpasswd -a name` 可完成此任务。使用以下命令为计算机创建 Windows 域概念要求的域帐户：

例 23.2 设置计算机帐户

```
useradd hostname\${
smbpasswd -a -m hostname
```

使用 `useradd` 命令可添加一个美元符号。命令 `smbpasswd` 在使用参数 `-m` 时自动插入此符号。带注释的配置示例 (`/usr/share/doc/packages/Samba/examples/smb.conf.SuSE`) 包含自动执行此任务的设置。

例 23.3 计算机帐户的自动设置

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \  
-s /bin/false %m$
```

为了确保 Samba 可以正确执行此脚本，应选择具有所需管理员权限的 Samba 用户。为此，请选择一个用户并将其添加到 `ntadmin` 组。然后可以使用以下命令来为属于此 Linux 组的所有用户指派 Domain Admin 状态：

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

有关此主题的详细信息，请参见位于 `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf` 的 Samba HOWTO 文档集的第 12 章。

23.5 有关详细信息

关于 Samba 的详细信息，请参见数字文档。在命令行输入 `apropossamba` 可显示一些手册页；如果安装了 Samba 文档，也可以浏览 `/usr/share/doc/packages/samba` 目录获得更多的联机文档和示例。可以在 `examples` 子目录中找到带注释的示例配置 (`smb.conf.SuSE`)。

Samba 开发小组提供的 Samba HOWTO 文档集中有一节专门介绍查错。此外，文档的第 V 部分提供了检查配置的逐步指南。安装包 `samba-doc` 后，可在 `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf` 中找到 Samba HOWTO 集。

另请阅读 openSUSE wiki 中的 Samba 页面，地址是 <http://en.opensuse.org/Samba>。

通过 NFS 共享文件系统

在企业环境中通过网络分发和共享文件系统是一项常见任务。NFS 是经过充分证实的系统，还可以与黄页协议 NIS 系统协同使用。要使用可以与 LDAP 协同使用、并且 Kerberos 化的更安全的协议，请选中 NFSv4。

NFS 与 NIS 一起使用以使网络在面向用户时是透明的。利用 NFS，可以通过网络分发任意文件系统。进行适当的设置后，用户将发现自己始终处于同一环境中，而与当前使用的终端无关。

24.1 安装所需软件

要将主机配置为 NFS 客户机，无需安装其他软件。配置 NFS 客户机所需的所有包都将默认安装。

24.2 使用 YaST 导入文件系统

经过授权的用户都可以将 NFS 服务器中的 NFS 目录装入自己的文件目录树。使用 YaST 的 *NFS 客户程序* 模块可以完成上述操作。单击添加并输入 NFS 服务器的主机名、要导入的目录以及在本地装入此目录的安装点。在第一个对话框中单击完成后，这些更改即会生效。

在 *NFS 设置* 选项卡中，单击 *打开防火墙中的端口* 打开防火墙，以便访问远程计算机上的服务。防火墙状态将显示在复选框旁边。使用 NFSv4 时，请确保启用

了启用 *NFSv4* 复选框，并且 *NFSv4* 域名包含与 *NFSv4* 服务器所用相同的值。默认域为 `localdomain`。

单击完成来保存更改。请参见图 24.1 “使用 YaST 配置 NFS 客户机” [272]。

配置写入 `/etc/fstab`，并将装入指定的文件系统。当您稍后启动 YaST 配置客户程序时，它还将读取此文件中的现有配置。

图 24.1 使用 YaST 配置 NFS 客户机



24.3 手动导入文件系统

还可以从 NFS 服务器手动导入文件系统。前提条件是要运行 RPC 端口映射器，以 `root` 身份输入 `rcrpcbind start` 即可启动它。一旦满足了这个前提条

件，通过以下方式使用 `mount` 命令，可以在文件系统中象装入本地硬盘那样装入远程导出的文件系统。

```
mount host:remote-path local-path
```

例如，如果应该导入某台计算机（如 `nfs.example.com`）上的用户目录，请使用以下命令：

```
mount nfs.example.com:/home /home
```

24.3.1 使用自动装入服务

除了装入通常的本地设备，`autofs` 守护程序还可以用于自动安装远程文件系统。要执行此操作，请在 `/etc/auto.master` 文件中添加以下条目：

```
/nfsmounts /etc/auto.nfs
```

如果 `auto.nfs` 文件正确完成，`/nfsmounts` 目录将作为客户机上所有 NFS 装入的根目录。文件名为 `auto.nfs` 是为了方便，也可以选择其他名称。在选定的文件（如果没有的话，就创建一个）中，如以下示例所示，添加所有 NFS 装入的条目：

```
localdata -fstype=nfs server1:/data  
nfs4mount -fstype=nfs4 server2:/
```

用 `rcautofs start` 激活设置。对于此示例，`/nfsmounts/localdata`，`server1` 的 `/data` 目录将通过 NFS 装入，`server2` 的 `/nfsmounts/nfs4mount` 将通过 NFSv4 装入。

如果在运行 `autofs` 服务时编辑 `/etc/auto.master` 文件，则必须重新启动自动装入程序才能使更改生效。请用 `rcautofs restart` 执行此操作。

24.3.2 手动编辑 `/etc/fstab`

通常，`/etc/fstab` 中的 NFSv3 装入项如下：

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

也可以手动将 NFSv4 装入添加到 `/etc/fstab` 文件中。对于这些装入，请在第三列中使用 `nfs4` 而不是 `nfs`，并确保在第一列中的 `nfs.example.com`：后面用 `/` 指定远程文件系统。`/etc/fstab` 中 NFSv4 装入的示例行如下所示：

```
nfs.example.com:/ /local/pathv4 nfs4 rw,noauto 0 0
```

`noauto` 选项可禁止在启动时自动装入文件系统。如果您要手动装入各个文件系统，则可以缩短用于装入的命令并且仅需要提供安装点，如下所示：

```
mount /local/path
```

请注意，如果您没有输入 `noauto` 选项，系统的初始化脚本将在启动时处理这些文件系统的装入。

24.4 采用 Kerberos 的 NFS

要对 NFS 使用 Kerberos 身份验证，必须启用 GSS 安全性。要执行此操作，请在初始 YaST 对话框中选择启用 GSS 安全性。请注意，您必须具有一个正在工作的 Kerberos 服务器才能使用此功能。YaST 没有设置服务器，但仅使用所提供的功能。如果希望使用 Kerberos 进行身份验证，则除了 YaST 配置外，还必须首先至少完成以下步骤，才能运行 NFS 配置：

- 1 请确保服务器和客户端都在同一 Kerberos 域中。这意味着它们访问相同的 KDC（密钥分发中心）服务器并共享其 `krb5.keytab` 文件（在任何计算机上的默认位置是 `/etc/krb5.keytab`）。
- 2 在客户机上用 `rcgssd start` 启动 `gssd` 服务。

关于配置采用 Kerberos 的 NFS 的更多信息，请参见[第 24.5 节“更多信息”](#) [274] 中的链接。

24.5 更多信息

除了 `exports`、`nfs` 和 `mount` 的手册页外，还可在 `usr/share/doc/packages/nfsidmap/README` 找到关于配置 NFS 服务器和客户端的信息。可在以下 Web 文档中找到联机文档：

- 在 SourceForge [<http://nfs.sourceforge.net/>] 上联机查找详细的技术文档。
- 关于设置采用 Kerberos 的 NFS 的描述，请参见 NFS Version 4 Open Source Reference Implementation [<http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html>].
- 如果您对 NFSv4 有疑问，请参见 Linux NFSv4 Frequently Asked Questions [<http://www.citi.umich.edu/projects/nfsv4/linux/faq/>] 常见问题解答。

文件同步

现今有很多人都在同时使用多台计算机 — 一台在家用，一台或多台在办公室用，还可能携带便携式计算机或 PDA 在路上用。很多文件是所有这些计算机上共同需要的。所以，您可能希望能在所有计算机上工作，修改文件，让所有计算机都能提供最新的数据。

25.1 可用的数据同步软件

数据同步对于通过快速网络永久互联的计算机而言并不是个问题。在这种情况下，使用 NFS 这样的网络文件系统并将文件储存在服务器上，就可以支持所有主机通过网络访问相同的数据。但如果网络连接较差或者不是永久连接，这种方法就行不通了。使用便携式计算机在途中工作时，所有所需文件的副本都必须位于本地硬盘上。不过，您需要随后同步修改的文件。在一台计算机上修改某个文件后，一定要更新该文件在所有其他计算机上的副本。对于零星的副本，可以用 `scp` 或 `rsync` 手动更新。但如果涉及大量文件，这个过程要复杂得多，您必须小心操作，避免出现旧文件覆盖新文件之类的错误。

警告：数据丢失风险

开始通过同步系统管理数据之前，您应该熟悉所用的程序并测试其功能。一定要对重要文件进行备份。

使用程序可以通过各种方法自动执行数据同步，从而克服手动同步数据时既耗时又容易出错的缺点。以下概要的目的只是让您大致了解这些程序的工作原理及它们的用法。如果打算使用它们，请阅读相应的程序文档。

25.1.1 CVS

CVS 主要用于对程序源代码进行版本管理；使用它可以在多台计算机上保留文件的副本。因此，该程序也适用于数据同步。CVS 在服务器上维护一个中央安装源，其中保存着文件和对文件的更改。本地执行的更改将提交到该安装源，并能够通过更新从其他计算机检索。这两个过程都必须由用户启动。

若多台计算机上都发生了更改，CVS 能够非常灵活地处理错误。这些更改将被合并，若发生在同一行上，则会报告冲突。发生冲突时，数据库仍保持一致状态。冲突仅显示在客户机上并在客户机上解决。

25.1.2 rsync

在无需版本控制但需要通过慢速网络连接同步大型目录结构时，rsync 工具可以提供较为完善的机制，仅传送文件中的更改。其中不仅涉及文本文件，还包括二进制文件。为检测文件间的差异，rsync 会将文件划分为多个块，并计算各个块的校验和。

检测更改需要消耗一定的资源。要使用 rsync，准备同步的系统应能够伸缩自如。RAM 尤为关键。

25.2 选择程序时的决定性因素

在决定使用哪个程序时请考虑几个重要因素。

25.2.1 客户机/服务器与对等模式

在分发数据时，常用的有两个模型。在第一个模型中，所有客户机都通过中央服务器来同步文件。所有客户机都应能够访问该服务器（至少能偶尔为之）。CVS 使用该模型。

另一个模型是让所有联网主机作为同级相互同步数据。rsync 实际在客户机模式下工作，但任何客户机都可用作服务器。

25.2.2 可移植性

CVS 和 rsync 还适用于其他很多操作系统，包括各种 Unix 和 Windows 系统。

25.2.3 交互与自动

在 CVS 中，数据同步是由用户手动启动的。这样可以有效控制要同步的数据并易于解决冲突。不过，如果同步间隔过长，就容易发生冲突。

25.2.4 冲突：事件和解决方案

在 CVS 中很少发生冲突，即便是多人同时在一个大型程序项目上协作时也不例外。这是因为合并文档时基于的是单个行。发生冲突时，只有一个客户机会受影响。通常很容易解决 CVS 中发生的冲突。

rsync 中不提供冲突解决功能。用户自己要避免意外覆盖文件，并手动解决所有可能的冲突。为安全起见，还可以使用 RCS 之类的版本控制系统。

25.2.5 选择和添加文件

在 CVS 中，必须使用命令 `cvsadd` 明确添加新目录和文件。这样用户可以更有效地控制要同步的文件。但另一方面，这样也容易遗漏新文件，特别是在有大量文件时，很容易忽略 `cvs update` 输出中的问号。

25.2.6 历史

CVS 的另一个功能是能够重建旧文件版本。每次一有更改都可以插入一个简短的编辑注释，以后根据文件内容和这些注释就很容易跟踪文件的变化。这对论文和程序文本大有帮助。

25.2.7 数据量和硬盘要求

所有相关主机的硬盘上都要有足够的可用于所有分发数据的空间。CVS 还要求服务器为安装源准备额外的空间。文件历史记录也储存在服务器上，这进一步

增加了空间要求。更改文本格式的文件时，只需保存修改的那些行。而二进制文件则要求在每次更改文件时都要有与文件大小相同的额外空间。

25.2.8 GUI

有经验的用户通常从命令行运行CVS。不过，图形用户界面也适用于Linux（如cervisia）以及其他操作系统（如wincvs）。许多开发工具（如kdevelop）及文本编辑器（如Emacs）都提供针对CVS的支持。在这些前端上解决冲突往往较为容易。

25.2.9 用户友好

rsync相当容易使用，还适合初学者。CVS某种程度上较难操作。用户应该了解安装源和本地数据之间如何交互。对数据的更改首先要在本地与安装源合并。使用命令 `cvs update` 可完成上述操作。然后必须使用命令 `cvs commit` 将数据发回安装源。一旦了解了此过程，新手也就能毫不费力地使用CVS了。

25.2.10 预防攻击

在传送数据的过程中，最好防止数据被拦截或操纵。CVS和rsync可以方便地通过ssh（安全shell）使用，从而防止遭受此类攻击。应避免通过rsh（远程shell）运行CVS。也不建议在不安全的网络中使用*pserver*机制访问CVS。

25.2.11 防止数据丢失

开发人员使用CVS来管理程序项目已有很长时间，所以该程序极为稳定。由于能够保存开发历史记录，CVS甚至能够预防某些用户错误，如意外删除文件。

表 25.1 文件同步工具的功能：-- = 很差，- = 差或不可用，o = 中等，+ = 好，++ = 很棒，x = 可用

	CVS	rsync
客户机/服务器	客户机-服务器	客户机-服务器

	CVS	rsync
可移植性	Lin、Un*x、Win	Lin、Un*x、Win
交互能力	x	x
速度	o	+
冲突	++	o
文件选择	所选/文件、目录	目录
历史	x	-
硬盘空间	--	o
GUI	o	-
难易程度	o	+
攻击	+(ssh)	+(ssh)
数据丢失	++	+

25.3 CVS 简介

如果经常编辑各个文件并且这些文件以 ASCII 文本或程序源代码文本之类的格式储存，则应该使用 CVS 来进行同步。用 CVS 同步其他格式的数据（如 JPEG 文件）固然可行，但这会产生大量数据，因为文件的所有变化都永久储存在 CVS 服务器中。这种情况下将无法利用 CVS 的大多数功能。只有在所有工作站都可以访问同一服务器时，才能使用 CVS 同步文件。

25.3.1 配置 CVS 服务器

服务器是储存所有有效文件（包括所有文件的最新版本）的主机。任何固定的工作站都可以充当服务器。如果可能，应该对 CVS 安装源的数据进行定期备份。

配置 CVS 服务器时，通过 SSH 授予用户访问服务器的权限是一种不错的方式。如果用户在服务器上的用户名为 `tux`，并且在服务器和客户机上都安装了 CVS 软件，则必须在客户端设置以下环境变量：

```
CVS_RSH=ssh CVSROOT=tux@server:/serverdir
```

可使用命令 `cvsinit` 从客户端初始化 CVS 服务器。只需执行一次初始化。

最后，必须给同步指派名称。仅在客户机上选择或创建目录，以包含要使用 CVS 来管理的文件（该目录也可以为空）。目录的名称同时也是同步的名称。在本例中，目录名为 `synchome`。转到此目录并输入以下命令，将同步名称设置为 `synchome`：

```
cvs import synchome tux wilber
```

许多 CVS 命令都需要注释。为此，CVS 会启动一个编辑器（在环境变量 `$EDITOR` 中定义的编辑器；如果未定义任何编辑器，则使用 `vi`）。通过提前在命令行中输入注释（如下例所示），可以避免调用编辑器。

```
cvs import -m 'this is a test' synchome tux wilber
```

25.3.2 使用 CVS

现在，在所有主机上都可以使用 `cvsco synchome` 将该同步安装源签出。该操作将在客户机上创建新的子目录 `synchome`。要向服务器提交更改，请转到目录 `synchome`（或其子目录之一），然后输入 `cvscommit`。

默认情况下，所有文件（包括子目录）都要提交给服务器。若仅提交单个文件或目录，请按 `cvscommit file1 directory1` 中的方式进行指定。在将新文件和目录提交给服务器之前，必须使用 `cvsadd file1 directory1` 之类

的命令先将其添加到安装源中。随后再使用 `cvscscommit file1 directory1` 命令提交新添加的文件和目录。

如果转到另一个工作站，则需要签出同步安装源（如果在同一工作站上的较早会话中尚未执行该操作）。

使用 `cvsupupdate` 开始与服务器同步。如 `cvsupupdate file1 directory1` 所示更新各个文件或目录。要查看当前文件与服务器上储存的版本的差异，请使用命令 `cvsdiff` 或 `cvsdiff file1 directory1`。使用 `cvs-nq update` 可以查看哪些文件将受到更新的影响。

以下是更新期间显示的一些状态符号：

U

已更新本地版本。这将影响服务器提供的和本地系统缺少的所有文件。

M

已修改本地版本。若服务器发生更改，可以将差异并入本地副本。

P

已使用服务器上的版本修补本地版本。

C

本地文件与安装源中的当前版本冲突。

?

此文件在 CVS 中不存在。

状态 M 表示本地修改的文件。可以向服务器提交本地副本，也可以在删除本地文件后再次进行更新。更新后将能够从服务器中恢复缺失的文件。如果提交了本地修改的文件但提交的这个文件中的同一行发生了更改，则可能发生冲突（由 C 表示）。

在这种情况下，查看文件中的冲突标记（“>>”和“<<”），决定要采用哪个版本。由于这是一项令人不快的工作，您可以选择放弃更改，删除本地文件，然后输入 `cvsup` 从服务器恢复当前版本。

25.4 rsync 简介

如果需要定期传送大量数据而更改的数据量不是很大，则适用 rsync。举例来说，创建备份时的情况往往就是这样。另一种应用涉及临时服务器。临时服务器是储存 Web 服务器的完整目录树的服务器，这些 Web 服务器定期镜像到 DMZ 中的 Web 服务器。

25.4.1 配置和操作

rsync 有两种操作方式。可用于存档或复制数据。要执行上述操作，目标系统上只需要有远程 shell，如 ssh。不过，rsync 也可用作守护程序，为网络提供目录。

rsync 的基本操作方式不需要任何特殊配置。rsync 能直接将完整目录镜像到其他系统中。举例来说，以下命令在名为 sun 的备份服务器上为 tux 的主目录创建了备份副本。

```
rsync -baz -e ssh /home/tux/ tux@sun:backup
```

以下命令用于回放该目录：

```
rsync -az -e ssh tux@sun:backup /home/tux/
```

到目前为止，该程序的操作方式与普通的复制工具 (如 scp) 的操作方式相差无几。

应该以“rsync”方式操作 rsync，以便充分利用其所有功能。这需要在其中一个系统上启动 rsyncd 守护程序。在文件 /etc/rsyncd.conf 中配置该守护程序。例如，要使目录 /srv/ftp 可用于 rsync，请使用以下配置：

```
gid = nobody
uid = nobody
read only = true
use chroot = no
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
```

```
[FTP]
```

```
path = /srv/ftp
comment = An Example
```


然后使用 `rcrsyncdstart` 启动 `rsyncd`。`rsyncd` 也可以在引导进程中自动启动。通过在 YaST 提供的运行级别编辑器中激活此服务或通过手动输入命令 `insservrsyncd`，都可以完成上述设置。也可以使用 `xinetd` 来启动 `rsyncd`。不过，建议只在很少使用 `rsyncd` 的服务器上采用这种启动方式。

下例还创建了一个列出所有连接的日志文件。此文件储存在 `/var/log/rsyncd.log` 中。

随后可以从客户机系统测试传送。请使用以下命令完成该操作：

```
rsync -avz sun::FTP
```

此命令列出服务器的 `/srv/ftp` 目录中现有的所有文件。此请求还记录在日志文件 `/var/log/rsyncd.log` 中。要启动实际的传送，请提供目标目录。使用 `.` 表示当前目录。例如：

```
rsync -avz sun::FTP .
```

默认情况下，使用 `rsync` 同步时不会删除任何文件。如果应强制删除，必须明确指定附加选项 `--delete`。为保证不删除任何较新的文件，可转而使用选项 `--update`。必须手动解决所有冲突。

25.5 有关详细信息

CVS

关于 CVS 的重要信息可以在主页 <http://www.cvshome.org> 中找到。

rsync

有关 `rsync` 的重要信息，请参见手册页 `manrsync` 和 `manrsyncd.conf`。`/usr/share/doc/packages/rsync/tech_report.ps` 专门提供了关于 `rsync` 工作原理的技术参考。在 `rsync` 的 Web 站点 <http://rsync.samba.org/> 上可以找到关于该项目的最新消息。

